ISCTEM — Instituto Superior de Ciências e Tecnologias de Moçambique
Universidade do Minho, Portugal

REPLICAÇÃO DE DADOS E SERVIÇOS

JOÃO FILIPE MACUACUA

Licenciado em Ensino de Matemática e Física Pela Universidade
Pedagógica em Maputo, 1991

Tese de Mestrado submetido para satisfação dos requisitos do grau Académico
de Mestre em Sistemas de Informação pela Universidade do Minho, e realizada sob a
supervisão científica do
Prof. Doutor Eng.º. Francisco Soares de Moura, DI-UM

Maputo, Fevereiro 2003

Resumo

Actualmente a utilização das Tecnologias de Informação (TI) nas organizações como suporte do negócio aumentou e todos os processos do negócio dependem da infra-estrutura das TI. O resultado da dependência do negócio em relação as TI é o facto de a partir de uma certa altura, a disponibilidade das aplicações passou a ser um elemento crítico para o sucesso do negócio. Por exemplo, se por um lado os empregados, clientes e fornecedores comunicam entre si e fazem trocas comerciais através das redes de dados, por outro lado uma falha no sistema quebra as relações estabelecidas e tem um impacto nos processos do negócio. Por essa razão o sistema de gestão de bases de dados passou a ser uma componente vital nas organizações. Na situação das empresas cujos rendimentos fluem através de *software* uma falha da base de dados pode custar milhões de *dólares* em cada hora.

Para evitar ou minimizar riscos e aumentar a confiabilidade e disponibilidade dos SI, muitas organizações renovaram as suas TI em relação à tolerância a falhas das bases de dados e de componentes, tendo esta renovação atingido maior relevância no final de 1999 com o problema da passagem do ano (ano 2000) e agravado com os acontecimentos de 11 de Setembro nos EUA.

Uma solução comum estrategicamente adoptada pelas organizações é a replicação de *hardware* ou de *software* em outros locais. Para os cenários de desastres que implicam a perda de dados e do lugar, a estratégia de replicação de serviços da base de dados incluem uma computação secundária num local remoto a qual permite que a danificação da fonte primária da base de dados não inviabilize todo o processo do negócio.

Como qualquer outra tecnologia, a replicação deve ser cuidadosamente compreendida antes de ser adoptada como solução numa organização, porque apesar de ter um custo efectivo na disponibilidade e confiabilidade, aumenta os desafios pelo número de locais participantes e a necessidade de consistência e eficiência.

O presente trabalho, examina as técnicas de replicação para a tolerância a falhas, disponibilidade e desempenho, considerando um caso de estudo onde apresenta propostas de como resolver problemas de tolerância a falhas e recuperação de desastres incluindo o bom desempenho num SI utilizando a replicação. A solução de destaque para o caso considerado

consiste na replicação passiva em dois centros de dados geograficamente distintos e a utilização da tecnologia *Storage Area Network (SAN)* nos dispositivos de armazenamento de dados que facilita a replicação.

Abstract

At this moment the use of Information Technology (IT) in organisations to support the business has increased dramatically with all business processes now depending on an IT infra-structure. The reason for the business depending on IT is due to the fact that from a certain time onwards, the availability of applications became a critical element in the success of the business. For example, if on one hand employees, customers, and partners communicate and conduct commerce through networked systems, on the other hand, when one system in the network fails, it can break dependencies and impact on business processes. For this reason the database management system has now become a vital component within organisations. In companies whose revenue flows through the software, database downtime can cost millions of dollars in lost business every hour.

To avoid or minimise risks and increase the reliability and availability of Information Systems (IS) many organisations upgraded their IT especially with regard to the fault-tolerance of databases and components. This upgrading reached its peak at the end of 1999 with the problem of stepping the year to 2000 (Y2K) and became all the more serious with the happenings on the 11th of September in the USA.

One common strategic solution adopted by organisations is the replication of hardware or software in other locations. For disaster scenarios that imply the loss of data or locations, the strategy of replication of database services include a secondary computing facility at a remote site which permits that the destruction of the primary database site will not neutralise the complete business process.

As with any other technology, replication must be carefully understood before being adopted as a solution in an organisation, because even though it is cost effective in terms of reliability and availability, it increases demands due to the number of locations and the necessity of consistency and efficiency.

The present paper examines in detail replication techniques for fault-tolerance, availability and performance, considering a study case where proposals on how to solve fault-tolerance and disaster recovery problems, including the improved performance of an IS using replication are presented. The preferred solution for the case considered consists of passive

replication in two geographically distinct data centres and the use of Storage Area Network (SAN) in the data storage depositories that permit the replication.

Agradecimento

O presente trabalho é o culminar de um processo longo de formação em Sistemas de Informação. A sua realização foi o resultado de muitas contribuições valiosas algumas das quais merecem um grande louvor.

Nesta perspectiva, e em primeiro lugar gostaria de endereçar os meus sinceros agradecimentos ao Conselho de Administração da empresa Telecomunicações de Moçambique, em especial à Direcção dos Sistemas de Informação com particular consideração ao Eng.º. Alfredo Borges, por ter apoiado a iniciativa e ter dado a oportunidade de estudar.

Em segundo lugar, pretende endereçar sinceros agradecimentos ao meu orientador Científico o Prof. Eng.º Doutor Francisco Soares de Moura pelo seu valioso conhecimento que se dignou generosamente partilhar.

Um apreço à Universidade do Minho, ao departamento dos Sistemas de Informação e em especial ao Departamento de Informática pelo grande apoio em material de investigação que concederam.

Grande consideração e reconhecimento aos Doutores Henrique Santos, Luis Amaral e o Professor Carlos Lauchande pela sua disposição e vontade na condução da dissertação.

Ao ISCTEM, vai uma grande retribuição para todos aqueles que de uma forma directa ou indirectamente apoiaram na realização deste trabalho.

Aos meus familiares um especial agradecimento, em particular a Adelina e aos meus filhos pela generosa solidariedade manifestada durante este percurso.

Aos meus amigos Nelson Chacha, Benedito Manguele e ao Consultor da Cornastone (A. do Sul) Sándor Vas endereço uma grande retribuição pela valiosa colaboração durante o trabalho.

Finalmente, agradece a todos os colaboradores da Direcção dos Sistemas de Informação da TDM com particular atenção aos colegas dos departamento de *Hardware* e Comunicações, *Software* de base e Base de dados, e a área das Telecomunicações da Beira pelo grande apoio e compreensão.

ÍNDICE

Índice de Figuras

Índice de Tabelas

Acrónimos e Abreviaturas

2PC	*Two-Phase-Commit*
2PL	*Two-Phase-Locking*
ACID	Atomicidade, Consistência, Isolamento e Durabilidade
API	*Application Program Interface*
ASE	*Adaptive Server Enterprise*
BD	Base de Dados
CICS	*Customer Information Control System*
DAS	*Direct-Attached Storage*
DBMS	*Database Management System*
DBA	*Database Administrator*
DCE	*Distributed Computation Environment*
DDL	*Data Description Language/Data Definition Language*
DML	*Data Manipulation Language*
DSS	Sistemas de Suporte à Decisão
FC	*Fibre Channel*
FTP	*File Transfer Protocol*
GUI	*Graphical User* Interface
ID	IDentificação
IDC	*International Data Corp*
KB	*KiloByte*
LAN	*Local Area Network*
LTM	*Log Transfer Manager*
MB	MegaByte
NAS	*Network Area Storage*
NIC	*Network Interface Card*
OF	*Oracle Financial*
OLAP	*Online Analytic Processing*
OLTP	*Online Transaction Processing*

RAID	*Redundant Arrays of Inexpensive Disks*
ROWID	*Row Identification*
ROWA	*Read One Write All*
RPC	*Remote Procedure Call*
RQS	*Recoverable Queuing Service*
RS	*Replication Server*
SFS	*Structure File Server*
SQL	*Structure Query Language*
QC	*Quorum Consensus*
SAN	*Storage Area Network*
SCSI	*Small Computer System Interface*
SGBD	Sistema de Gestão da Base de Dados
SI	Sistemas de Informação
TI	Tecnologias de Informação
TDM	Telecomunicações de Moçambique
TP	*Transaction Processing*
UID	*User Identification*
UPS	*Uninterruptible Power Suplier*
WAN	*Wide Area Network*

Capítulo I

1. Introdução

A arquitectura cliente/servidor e a correspondente centralização dos Sistemas de Informação (SI) em servidores acedidos remotamente permitem as organizações soluções informáticas e baixo custo, principalmente na aquisição e manutenção, mas tornam muitos serviços dependentes do acesso remoto. A disponibilidade do serviço passa a depender da disponibilidade do servidor e das comunicações. Como consequência, nas organizações com dados críticos onde não se tolera a falta do serviço há uma maior tendência para o emprego de redundância dos servidores e de outros recursos como forma de aumentar a tolerância a falhas e disponibilidade.

Uma das formas de implementar a tolerância a falhas é através do uso de vários servidores independentes. O estado dos serviços é replicado e distribuído nesses servidores e a actualização é coordenada em todos os servidores, para permitir que a falha de uma parte destes não implique necessariamente a indisponibilidade do serviço. No que respeita ao *software*, nos sistemas distribuídos ele é estruturado em função dos clientes e serviços. Cada serviço compreende um ou mais servidores e exporta operações às quais os clientes acedem quando fazem o pedido. Em geral, as réplicas de um serviço simples são executadas em processadores independentes, e protocolos de sincronização são empregues para coordenar a interacção do cliente com as réplicas. O isolamento físico e electrónico dos processadores garantem a independência das falhas.

Existem diferentes maneiras de implementar a replicação, sendo frequente encontrar duas classes: a primeira é designada por replicação activa ou metodologia da máquina do estado; e a segunda é designada por primário/*standby* ou replicação passiva. Nos casos mais simples a replicação é implementada usando os *backups* em bandas magnéticas (*tapes)* os quais são uma garantia de reposição de dados ou de serviços em caso de falhas. Este método tem o inconveniente de ter os dados em *offline* e de guardar sempre dados atrasados, além de requerer muito mais tempo para repor o estado mais recente dos dados. Já nos casos mais sofisticados, a replicação procurará a manutenção *online* de cópias de dados ou outro tipo de recursos para

garantir a tolerância a falhas, alta disponibilidade e um bom desempenho [5].

Os factores que condicionam a utilização da replicação são a redução de custos em relação a transferência de dados, a redução do tempo de resposta na execução de uma transacção e a prevenção contra falhas, principalmente nos dados críticos de uma empresa [10].

1.2. Modelo Centralizado e Distribuído de Dados nas Organizações

A crescente dependência do negócio das empresas nas Tecnologias de Informação (TI) aumentou a vulnerabilidade económica associada a falhas destas tecnologias. Como consequência muitas empresas estão a construir uma infra-estrutura de suporte segura que minimize as perdas. Nenhuma empresa quer implementar mecanismos tradicionais em que uma falha possa demorar três a quatro dias para a sua recuperação. Os Sistemas de Informação actuais, incluindo *E-commerce, Data-Warehousing* e *data-mining,* frequentemente exigem uma disponibilidade total e pouco tempo de recuperação, para além de um bom desempenho. Uma recuperação da base de dados em pouco tempo exige a construção de uma arquitectura de replicação de dados dentro do seu plano de recuperação de desastres em outros locais alternativos – distribuição de dados. A solução de replicar os dados é específica para uma base de dados, *file systems*, sistemas operativos ou subsistemas de discos. Para a protecção dos dados críticos as empresas devem utilizar soluções múltiplas incluindo métodos de segurança do meio envolvente à replicação.

O modelo centralizado de dados não inclui a distribuição de dados, todos os dados estão guardados num único gestor de recursos. A utilização de dados a nível da empresa é distribuída e feita via acesso remoto para várias unidades funcionais. Este modelo oferece melhor facilidade na consistência de dados porque todos os utilizadores vêm o mesmo estado global da base de dados em qualquer instante. Para além disso, o modelo centralizado torna simples os mecanismos de controlo, segurança e manutenção de dados. Todavia, um sistema centralizado caracteriza-se pela existência de uma cópia de dados constituindo o único ponto de falha. Quando ocorre uma falha na fonte de dados que torna o acesso indisponível todo o sistema pára de funcionar. Para além disso o modelo implica um elevado potencial de utilização da rede de comunicação de dados, resultando em custos elevados para a empresa.

Comparado com o sistema centralizado um sistema distribuído tem melhor

aproveitamento de recursos e teoricamente, tem maior fiabilidade, melhor gestão da redundância e melhor crescimento. Em contrapartida, tem *software* mais complexo com maiores dificuldades em termos de configuração e gestão.

1.3. Descrição geral do problema

O autor é colaborador da empresa TDM, Telecomunicações de Moçambique. Tal como outras empresas de grande dimensão, a TDM tem uma estrutura de carácter distribuído que contrasta com os seus Sistemas de Informação centralizados. Este caracter distribuído é visível pela descentralização das funções do negócio, procurando abranger um conjunto mais amplo de locais e clientes. Este facto origina uma forte dependência das infra-estruturas de suporte aos seus Sistemas de Informação. Em particular, o acesso a dados e serviços depende da disponibilidade da rede de comunicação de dados, o que condiciona a disponibilidade dos serviços. Acresce ainda em alguns casos a fraca capacidade tecnológica para resolver problemas de tolerância a faltas e desempenho das aplicações, conduzindo ao estrangulamento de trafego e menor disponibilidade dos serviços. Além disso, a centralização dos recursos num lugar pode ser prejudicial em termos de segurança de dados, em particular na prevenção de falhas catastróficas.

Existem já disponível soluções tecnológicas e mecanismos que podem ajudar as organizações e as instituições a melhorar a sua segurança e tolerarem as falhas, tanto de *hardware* como de *software*. Uma dessas alternativa é o armazenamento redundante dos dados em locais geograficamente distintos. Muitas empresas já adoptam soluções deste tipo, mas que, frequentemente, evidenciam limitações porque os servidores estão centralizados num local, e sem mecanismos de replicação que possam garantir a automatização da recuperação de falhas tornando esta medida administrativa. Uma outra alternativa consiste no armazenamento de dados de modo redundante, mas distribuindo-os pelos locais onde eles são necessários, e estabelecendo um mecanismo de sincronização, o que conduz a uma descentralização das unidades funcionais. Os vendedores de *hardware* e *software* fornecem soluções com estas características.

No caso da empresa pública TDM, onde a política do governo incentiva a concorrência, é cada vez mais necessário garantir a disponibilidade dos serviços e economizar o tempo. É por

esta razão que se torna importante o recurso a novos métodos e tecnologias que ajudam a empresa a responder adequadamente às necessidades dos seus clientes.

Os SI desempenham um papel fundamental no fluxo da Informação, substituindo os mecanismos tradicionais, reflectindo as suas vantagens no manuseamento de grande quantidade de informação em pouco tempo, para além de encurtar cadeias de processos complexos. Os SI são abertos e baseados na política de administração centralizada. As aplicações principais são a gestão do serviço telefónico fixo, a gestão financeira, a gestão dos recursos humanos e os serviços de acesso à Internet. A arquitectura utilizada é a cliente/servidor, em todos os casos. Na gestão do negócio são empregues dois servidores, um de produção e um outro de espera (*backup)*. Mas tem sido ao nível das comunicações que as maiores dificuldades tem surgido, obrigando as províncias a interromper os serviços durante longos períodos, tal como se mostra na tabela 1.1, com o registo das falhas de uma semana . São reportado falhas das comunicações devidas à falha do hardware envolvido (*hub e routers*).

Data	Província/local	Inícial (falha)	Final (falha)	Duração
10/11/2001	Inhambane	7:30 horas	11:00 horas	4:10 horas
11/11/2001	Inhambane	11:00 horas	12:30 horas	01:30 Horas
13/11/2001	Lichinga	10:30 horas	14:53 horas	03:53 Horas
16/11/2001	Todos os sites remotos	12:00 horas	18:00 horas	6:00 Horas

Tabela 1.1. Falhas da rede, Novembro 2001

No centro de dados, Maputo, a disponibilidade e fiabilidade é baixa devido a várias interrupções incluindo a falha da base de dados e falta dos serviços de energia e de telecomunicações. Na aplicação Girafe, a utilização do servidor de redundância não é eficaz requerendo medidas complementares de replicação. A centralização de dados sem mecanismos de distribuição resulta dentro da empresa num menor desempenho caracterizado por elevado tempo na recuperação das falhas e elevada latência nos pontos de acesso mais distantes. Um outro problema que afecta os SI da empresa relaciona-se com os procedimentos dos *backups* e planos de recuperação de desastres porque consomem muito mais tempo afectando o desempenho e a disponibilidade.

1.4. Objectivos de trabalho

O presente trabalho tem dois objectivos principais

1. Propor técnicas de replicação de dados e serviços que mais se adequam à TDM, por forma a aumentar a tolerância a falhas, desempenho e segurança de dados distribuídos.
2. Identificar e definir mecanismos que possam ser utilizados para a melhor confiabilidade, disponibilidade e o desempenho do SI da TDM.

1.5. Estruturação da dissertação

Este trabalho é constituído por duas partes, sendo a primeira parte teórica e a segunda correspondendo ao caso prático de estudo. O primeiro capítulo aborda a replicação no contexto geral as vantagens que traz para as organizações, as limitações dos modelos centralizados no tratamento da tolerância a falhas, disponibilidade e o desempenho. É tratada também a descrição dos problemas das organizações cuja solução pode ser o método da replicação e os objectivos do trabalho.

No capítulo dois e três faz-se a alusão aos aspectos teóricos sobre a tolerância a falhas, desempenho e alta disponibilidade. Especificamente no capitulo dois é abordada a replicação fundamentalmente para responder às questões de desempenho, alta disponibilidade e tolerância a falhas nas organizações. No capítulo três abordam-se as técnicas de replicação em pormenor, apresentam-se os modelos de posse de dados, mecanismos de consistência, protocolos ROWA e QC para a actualização dos dados replicados. Trata-se também a comunicação em grupo e explica-se a utilização das vistas para a transferência do estado de um processo ou servidor para o outro. No capítulo quatro faz-se uma abordagem geral dos fornecedores dos produtos de replicação e as estratégias de replicação que implementam.

A segunda parte deste trabalho trata o caso do estudo que consiste no levantamento dos Sistemas de Informação da TDM com grande enfoque para os problemas de tolerância a falha, desempenho e alta disponibilidade. É feita uma avaliação precisa das dependências funcionais relativa às Bases de dados e Internet. A discussão dos problemas surge no capítulo 6 assim como

as soluções propostas para os problemas levantados no capitulo anterior. Finalmente é feita uma análise geral ao trabalho realizado, destaque para as questões de plano de recuperação de dados tendo em consideração os diferentes métodos de armazenamento de dados, nomeadamente o *Direct-Attached Storage* (DAS), o *Network Area Storage* (NAS) e o *Storage Area Network* (NAS), ver o anexo B. No capítulo 7 são recapituladas as ideias principais do trabalho e apresentada propostas relevantes para melhorar o sistema em estudo.

Capítulo II

2. Replicação em Sistemas Distribuídos

2.1. Considerações Gerais sobre a Replicação

Os recursos físicos usados para armazenar dados são por sua própria natureza susceptíveis a falhas. Um conjunto elevado de falhas pode atingir os meios de armazenamento, quer sejam falhas no próprio meio, nos dispositivos de controle, nos servidores que fazem a gestão destes dispositivos, ou nos canais de comunicação. O uso de grandes redes institucionais e servidores corporativos, com dezenas ou centenas de utilizadores, tende a agravar este problema. Em sistemas de redes aumenta o número de pessoas ou actividades que dependem dos mesmos discos e servidores de dados/ficheiros, criando condições para que aconteçam perdas simultâneas ou em cascata relacionadas à falhas em algum dos servidores. As consequências de falhas podem ser muito graves e até catastróficas.

O aumento de velocidade das redes modernas permite a adopção de técnicas de replicação de arquivos e a distribuição de réplicas por vários servidores. A técnica de replicação é o centro de todo sistema de redundância de dados distribuídos, e determina de forma decisiva as opções a serem tomadas no tratamento de falhas e na respectiva recuperação.

2.2. Tipos de replicação

Um sistema diz-se assíncrono quando não faz nenhuma assunção acerca da velocidade de execução do processo e/ou em relação a entrega da mensagem. No caso contrário diz-se síncrono [9]. Um sistema síncrono faz assunção dos limites de execução e a demora dos canais de comunicação. Todos os sistemas são assíncronos por isso que são desenhados os protocolos para serem usados em tais sistemas que podem ser empregues também em sistemas distribuídos. Um canal de comunicação diz-se síncrono quando o emissor e o receptor de processo sincronizam em cada mensagem e bloqueiam as suas operações a espera da resposta [9]. A forma de comunicação

assíncrona não bloqueia as operações do emissor, no envio do processo, enquanto que o receptor pode ter ou não o bloqueio. Na variante de não bloqueio, a recepção do processo continua com o programa depois de uma operação de emissor. Um sistema de comunicação assíncrona que não bloqueia o cliente é relativamente eficiente, mas envolve complexidade adicional na recuperação de processos. Um canal de comunicação diz-se confiável quando está garantida a entrega das mensagens em relação a um número razoável de pacotes perdidos [9].

A replicação de dados diz-se síncrona quando os dados são sempre os mesmos em todos os gestores de recursos (uso de protocolos de *commit* global) e a latência é igual a zero. Uma replicação de dados diz-se assíncrona quando há sempre um atraso entre o *commit* da transacção originadora das modificações e o seu efeito ou disponibilização do seu efeito nas réplicas, isto é, quando a latência é superior a zero. Em termos comparativos, a replicação assíncrona reduz os custos e aumenta a concorrência nos gestores de recursos [5,9].

Replicação activa ou síncrona consiste na manutenção em funcionamento de dois ou mais gestores de recursos (servidores) em simultâneo. Caracteriza-se pela consistência rígida entre os vários gestores de recursos e pelo emprego de protocolos de *commit* global, habitualmente designados por *Two-Phase-Commit* (2PC). Replicação passiva ou primário/*backup* consiste na manutenção de dois gestores de recursos (servidores) em funcionamento mas, em cada instante um é primário e um ou mais estão em espera.

2.3. Perspectivas da replicação

A replicação pode ser empregue com vários propósitos, a maior parte dos quais resumem-se a:

- Elevar disponibilidade de dados e serviços
- Elevar a confiabilidade dos sistema informáticos
- Aumentar o desempenho
- Aumentar a tolerância a falhas e segurança de dados

2.3.1. Disponibilidade de dados e serviços

Um sistema diz-se disponível quando está operacional durante uma fracção de tempo t (tempo de emissão) [1,6,9].

Os factores críticos na disponibilidade de dados e serviços são:

- Falha do servidor
- Falha da partição da rede de comunicação de dados

O primeiro caso pode-se resolver usando a replicação dos serviços, isto é, *hardware* e *software* necessários à manutenção do serviço, em dois ou mais lugares com falhas independentes.

O segundo caso pode-se resolver empregando métodos de detecção e resolução de conflitos, por exemplo, entre utilizadores móveis que perdem temporariamente a consistência de dados. Este problema é posteriormente resolvido através de métodos de detecção e resolução de conflitos quando se estabelece a sincronização de dados com o servidor principal de dados [9].

Nas falhas da rede também pode-se utilizar adaptadores de falhas e um agente de *software* inteligente que continuamente vai monitorar os canais de comunicação. Na falha de uma componente do canal o outro redundante arranca imediatamente. Estes métodos incluem componentes de alta disponibilidade de aumentar a tolerância a falhas [*intel*].

Os métodos usados para avaliar a disponibilidade são três, dois dos quais baseados na probabilidade de falha *D(t)* de uma componente no intervalo de tempo definido [0,t].

O primeiro método é experimental e consiste nas observações dos estados do sistema durante o intervalo de tempo definido [0,t], registando-se tempos correspondente ao intervalo de tempo no qual o sistema está disponível *(Ui).*

A disponibilidade é dada por [1]:

$$D(t) = \frac{\sum_i Ui}{t}$$

O intervalo [0,t] deve ser escolhido com base nos tempos de utilização do sistema ou seja o chamado tempo de emissão.

O segundo método da avaliação da disponibilidade de um objecto (ficheiro ou dado) é

calculado como a diferença entre a unidade e a probabilidade da falha da componente ou gestor de recurso.

$D(t) = 1 - p^n$ Onde *n* é o número de componentes ou servidores que podem falhar.

No terceiro método a disponibilidade *D(t)* do sistema operacional num ponto arbitrário t, calcula-se como o limite da razão entre o Tempo Médio da Falha (TMF) e a soma do Tempo Médio da Falha e Tempo Médio de Reparação (TMR) quando o ponto t tende para infinito.

Quer dizer:

$$D(t) = \lim_{t-\infty} \frac{TMF}{TMF + TMR}$$

2.3.2. Confiabilidade dos sistemas informáticos

Fiabilidade – satisfação da especificação do serviço quando disponível. Fiabilidade do sistema refere-se às propriedades de tolerância a falha de componentes constituintes de um sistema, durante o tempo de emissão (ou de trabalho). Normalmente, para se implementar um sistema distribuído confiável é adicionada uma componente de *software* e/ou de *hardware* de tolerância a falha. Esta componente inclui protocolos de detecção de falhas, de recuperação e de reconfiguração depois da falha. A fiabilidade é um aspecto crítico nos sistemas informáticos porque inclui a disponibilidade e segurança. Além disso, a confiabilidade deve prevenir os acessos não autorizados, corrupção e inconsistência de dados.

A fiabilidade mede-se através da probabilidade *C(t)* de o sistema estar a funcionar correctamente durante o intervalo de tempo [0,t]. Se considerarmos f(x) uma função de densidade da falha podemos determinar a confiabilidade da seguinte maneira [17]:

$C(t) = 1 - \int_0^t f(x)dt = 1 - F(x)$ onde $f(x) = \alpha e^{-\alpha x}$

2.3.3. Desempenho

O emprego da replicação como forma de aumentar o desempenho nos sistemas distribuídos baseia-se na colocação das réplicas perto do utilizador ou grupo de utilizadores. O desempenho aumenta principalmente nos sistemas de grandes dimensões e com redes propensos a partição. Os dados partilhados num grupo de utilizadores não devem ser armazenados num único servidor porque podem aumentar o tempo de resposta (congestionamento).

As estratégias usadas para elevar o desempenho passam pela replicação de servidores e replicação de dados. A replicação de servidores é mais utilizada na gestão distribuída das transacções com réplicas localizadas perto de um grupo de utilizadores enquanto que a replicação de dados é muito utilizada na hierarquia da *cache* da *web* replicando as páginas no servidor *proxy* localizado perto do grupo de utilizadores diminuindo deste modo o tempo de resposta [9,13].

A técnica de replicar os servidores também pode ser utilizada para o equilíbrio da carga computacional e melhorar a qualidade dos serviços. Um sistema informático com acessos remotos o tempo de resposta é muitas vezes determinado pela carga do servidor e da rede. Para além disso o tempo de resposta é também determinado ou influenciado indirectamente pelo *software* interveniente, *middleware* sendo este código que implementa o serviço e a transferência de dados e o respectivo controlo entre os processos intervenientes. É por isso que nos sistemas com acesso remoto se pode melhorar o desempenho se o sistema for constituído por poucas camadas de *software* e uma menor quantidade de dados a serem transferidos entre o cliente e o servidor [9]. Por exemplo, a execução de um *applet* no computador do cliente retira carga ao servidor da *web*, oferecendo um melhor serviço. Um exemplo mais significativo verifica-se no uso de vários servidores para disponibilizar vários serviços simples.

O desempenho mede-se através de *throughput*, que indica a taxa de execução de trabalho num sistema. O *throughput* num sistema distribuído é afectado pela velocidade de processamento do cliente e servidor e pela taxa de transferência de dados.

2.3.4. Tolerância a falhas e segurança de dados

Tolerância a falha é a capacidade de um sistema continuar a funcionar mesmo quando ocorre uma falha num dos seus componentes. Um sistema que falha pode degradar o seu desempenho e piorar a sua confiabilidade. O objectivo dos sistemas tolerantes a falha é garantir que o sistema como um todo continue a funcionar correctamente, mesmo na presença de falhas.

Uma falha do sistema informático pode ser originada pela falha das suas componentes, nomeadamente *software*, *hardware* e erros humanos (ou desastres). Na prática cada uma das causas mencionadas poderá constituir um subgrupo de outras causas. Por exemplo, a falha de *software* pode ser causada pela falha de sistema operativo, ou pela falha da base de dados, da aplicação, da rede ou de *middleware* [23].

Uma componente diz-se que falha quando o seu comportamento não é consistente com as suas especificações [7]. Existem três tipos de falhas: transientes, intermitentes e permanentes. As falhas transientes ocorrem uma vez e desaparecem; as falhas intermitentes ocorrem de modo alternado; as falhas permanentes são aquelas que permanecem até que a componente que falhou seja reparado. No que respeita aos processadores, a classificação de falhas pode ser feita com mais algum detalhe, nomeadamente podem ser: *fail-stop* e *Byzantine*. Quando ocorre uma falha do tipo *fail-stop*, o processador pára de funcionar e deixa de dar resposta aos *inputs* subsequentes ou produz *outputs* a mais [6]. Quando ocorre uma falha do tipo *byzantine* o processador continua a funcionar, produzindo respostas erradas aos pedidos, dando impressão de estar a funcionar correctamente. Os erros de software não detectados evidenciam este tipo de falhas.

Em sistemas distribuídos com acesso remoto ambos os processos e canais de comunicações podem falhar. As falhas nas comunicações podem ser: de omissão, *byzantine* (ou arbitrárias) e de temporização. As falhas de omissão são aquelas que um processo ou canal de comunicação falha na execução de acções que é suposto a realizar. O outro processo detecta a falha pela ausência da resposta, isto é, utiliza a detecção de falhas baseada no método dos *timeouts* (um processo admite um período fixo de tempo para que alguma coisa ocorra). Num sistema assíncrono um *timeout* pode indicar somente que um processo não está a responder, podendo ter falhado ou ser lento, ou ainda a mensagem ainda não chegou. Um comportamento *fail-stop* pode ser produzido num ambiente dos sistemas síncronas quando o processo usa os

timeouts para detectar a falha do outro processo e se há alguma garantia de entrega. Uma falha arbitrária é aquela que omite arbitrariamente etapas de processamento pretendido ou toma etapas de processamento não pretendido. Todavia, as falhas arbitrárias não podem ser detectadas pela verificação de se o processo responde à invocação ou não, porque pode omitir arbitrariamente as respostas. Nos canais de comunicação uma falha arbitrária pode manifestar-se pela corrupção do conteúdo das mensagens ou adição de mensagens não existentes ou as mensagens verdadeiras serem entregues mais que uma vez. Devido a melhorias na tecnologia do *hardware* e *software* de comunicações esta classe das falhas é rara porque é possível detectar o erro e rejeitar as mensagens que falharam. Por exemplo, os *checksums* são usados para detectar mensagens corrompidas; o número de sequência das mensagens pode ser usado para detectar mensagens duplicadas ou não existentes.

As falhas de temporização ocorrem nos sistemas síncronos distribuídos onde os limites do tempo são fixados no tempo de execução do processo. Num sistema distribuído assíncrono, um servidor sobrecarregado pode responder muito lentamente, mas não se pode dizer que tenha uma falha de temporização desde que não sejam dadas quaisquer garantias. Os sistemas operativos em tempo real são desenhados com vista a oferecer garantias de temporização, mas o seu desenho é mais complexo requerem *hardware* redundante. A temporização é particularmente relevante para computação multimédia, com canais de áudio e vídeo.

Quando se usa redundância de dados como mecanismo de aumentar a tolerância a falha, as réplicas podem ser guardadas e sincronizadas de duas maneiras diferente, respectivamente: tempo real e próximo de tempo real.

a) Em tempo real

Quando se usa a sincronização das replicas em tempo real, alternativa que habitualmente é designada por *hot-standby*, a consistência total de dados é rígida e pode ser alcançada recorrendo-se a mecanismos de *software* e/ou de *hardware*.

Quando se emprega a técnica de *hardware* como alternativa de alcançar a consistência de dados, este deve proporcionar uma escrita dual em todos os dispositivos redundantes. No entanto, a consistência baseada no *hardware* tem duas desvantagens importantes. Em primeiro lugar,
os dois dispositivos devem estar próximos um do outro, para oferecer protecção nas falhas da

base de dados. Isto é prejudicial nas falhas locais, principalmente os desastres. Em segundo lugar, uma falha ou erro que origine a corrupção de dados no primário vai-se duplicar na réplica de *standby*.

Quando se usa *software* como alternativa de implementação das funcionalidades de *hot-standby* utiliza-se a replicação síncrona para manter a consistência entre as réplicas primárias e de *standby*. Esta metodologia não é transparente para o utilizador ou para as unidades distribuídas de trabalho, porque requer a modificação do código da aplicação para incluir as funcionalidades de tolerância a falha [19]. Devido ao acesso concorrente dos dados e/ou réplicas, a replicação *hot-standby* pode reduzir o desempenho da aplicação por causa dos *locks* em vários gestores de recursos e as comunicações, o que não acontece na replicação passiva.

b) Próximo de tempo real

Neste caso, o servidor em *standby* (*warm-standby)* utiliza uma replicação assíncrona que tem comparativamente ao *hot-standby* as seguintes vantagens:

- Pode ser implementado de modo transparente para as unidades distribuídas de trabalho.
- Não utiliza os *locks* concorrentes nos gestores de recursos [19].
- Não é necessário que os gestores de recurso ou fonte primária e de *standby* estejam perto um do outro, o que tolera todas as falhas incluindo acidentes naturais e sabotagens.

A mudança do primário para o *standby* deve ocorrer com o mínimo de preparação, podendo ser desencadeado por meio de um procedimento preestabelecido, obedecendo a uma quantidade de tempo prognosticável. Pode existir uma menor degradação do desempenho, dependendo da metodologia arquitectural usada na implementação.

Existem duas metodologias aceites na implementação de replicas *warm-standby.*

A primeira consiste na listagem dos *logs* da base de dados do primário na réplica *standby* – o *rolling database logs* propaga o *dumping* periódico dos *logs* das transacções na base de dados do primário e a subsequente aplicação destas no *standby* réplica. Mas o *rolling database logs* geralmente implica um maior tempo de latência nas réplicas de *standby,* o que origina uma replicação assíncrona contínua. Um maior tempo de latência dos dados implica um elevado potencial de perdas de transacções, quando ocorre uma falha do primário. O *rolling the log* pode

originar a duplicação de efeito de erro de *software* tornando a base de dados ou dispositivo corrompida. Durante o processo de aplicação dos *logs* na réplica *standby*, a base de dados permanece no modo de recuperação e por isso está indisponível para a aplicação dos utilizadores.

A segunda metodologia é o uso da replicação assíncrona – geralmente é usada a replicação assíncrona do tipo base de dados para base de dados. Com esta metodologia, existe uma menor probabilidade de corrupção da base de dados primária que possa ser transmitida para a réplica *standby*. O tempo de latência é menor, comparado com o do caso anterior. A desvantagem na implementação da replicação assíncrona reside na maior complexidade da implementação e respectiva manutenção.

2.4. A validação das técnicas de tolerância a falha

Uma dificuldade importante na implementação da tolerância a falhas é saber se as técnicas implementadas resultam realmente em aumento da confiabilidade. Como na maior parte dos Sistemas a taxa de falha é pequena e as falhas acontecem de forma aleatória, o problema resume-se em avaliar se a técnica empregada tolera a falha para a qual foi desenhada, sem a necessidade de esperar meses ou anos para que a falha aconteça.

As técnicas usadas para testar a confiabilidade em muitos sistemas é a injecção de falhas [20].

Através da introdução controlada de falhas, a eficácia da tolerância pode ser avaliada. A desvantagem desta técnica é que ela pode produzir efeitos indesejáveis que podem resultar na danificação de algum componente de *hardware* do sistema em teste. Por essa razão há uma tendência maior em fazer a injecção de falhas usando o *software,* é económica e tecnicamente viável porque é de fácil desenvolvimento. É facilmente adaptável a novas classes de falhas, embora, não modele todo o tipo de falhas, como por exemplo aquelas que afectam as unidades de controlo do processador.

2.5. Mecanismos de detecção e recuperação das falhas

Para garantir um bom funcionamento de um sistema distribuído replicado, é necessário ter mecanismos e técnicas que possam detectar a falha e manter a disponibilidade dos serviços. A detecção de falhas e métodos de recuperação baseiam-se em mecanismos de *hardware* e de *software*, para identificar a ocorrência da falha e corrigi-la de modo a garantir o funcionamento do sistema. Um conhecimento da característica da falha de um componente pode permitir a designação de um novo serviço para mascarar a falha por esconder ou converter num outro tipo de falhas mais aceitáveis. Por exemplo, os *checksums* são usados para mascarar as mensagens corrompidas, efectivamente, convertendo-as falhas arbitrárias nas falhas de omissão. As falhas de omissão podem ser mascaradas pelo uso de protocolos que re-transmitem as mensagens que não chegam aos destinatários; a replicação do processo é um outro método de mascarar as falhas que tem lugar quando um processo falhou.

A detecção de falhas nos sistemas síncronos replicados baseia-se num protocolo que periodicamente verifica o estado da componente em observação recebendo mensagens "eu estou vivo" emitidas pela componente em observação. Quando a mensagem do "pulsar do sistema"(eu estou vivo) não chega ao detector o processo ou componente monitorizado é considerado ter falhado. Os sistemas reais funcionam na base destes métodos mas com algumas optimizações. A detecção de falha considera-se perfeita quando o detector de falhas e a componente monitorizado estiverem perto um do outro porque o canal de interacção garante um certo grau de precisão na detecção do erro. Muitos sistemas de *hardware* e *software* incluem detectores de falhas dentro da componente monitorizado para diminuir a imperfeição do canal. Por exemplo, as rotinas de auto-verificação da paridade na memória, nos discos e nos *buses.* A desvantagem deste método verifica-se quando a monitorização não é baseado no *kernel* (unidade de controlo) porque limita a informação do detector de falha e cria dificuldades em identificar uma falha de facto da eventual lentidão do sistema. Um canal de interacção imperfeito pode causar falhas de omissão que muitas vezes pode ser resolvidas pela redundância do canal.

Os outros métodos de detecção e recuperação de falhas nos sistemas replicados incluem : transacções atómicas, servidor *stateless, aknowledgments (ack) and timeout-based retransmissions of messages,* replicação simétrica e o *Mirroring* [3,19].

Transacção atómica: considera cada transacção como um conjunto de operações indivisíveis. Uma transacção é um processo de tudo ou nada, isto é, ou as operações são todas executadas completamente com sucesso, *COMMIT*, ou são abortadas, *ROLLBACK*, mantendo o último estado consistente. As transacções podem preservar a consistência de um conjunto de dados de objectos partilhados, quando ocorre uma falha é mais fácil recuperar uma transacção (o seu estado anterior) porque tem apenas dois estados. Quando um sistema não tem facilidades de transacções atómicas torna difícil ou impossível fazer o recuo do estado de dados a partir de um estado inconsistente. Este é o método comum de recuperação de erro nos sistemas de bases de dados.

- Servidor *stateless*: o modelo cliente servidor é muito usado nos sistemas distribuídos. Neste modelo, um servidor pode ser implementado usando dois paradigmas de serviços: *stateless* ou *stateful*. Estes distinguem-se por um aspecto da relação cliente servidor, relacionado com o facto da historia de uma transacção efectuada entre o cliente e o servidor poder afectar (*stateful*) ou não (*stateless*) a execução da próxima transacção. O mecanismo *stateless* tem vantagem porque não guarda informação suplementar de cada transacção, sendo, por isso, muito mais fácil recuperar o seu estado em caso de falha.
- *Acknowledgments and timeout-based retransmissions of messages: n*os sistemas distribuídos as falhas nos locais onde residem os dados ou falhas nos canais de comunicação podem interromper a comunicação em curso entre dois processos, originando a perda de mensagens. Mas, um sistema de comunicação confiável deve oferecer algum mecanismo que permita detectar esta perda de mensagens e re-transmiti-las. Nesta perspectiva, a entrega e recepção de mensagens envolve outros mecanismos, como o *message acknowledgemente* e a re-transmissão baseada em *timeouts*. Isto é, conforme ilustrado na figura seguinte, o receptor deve retornar uma mensagem de *acknowledgement* para todas as mensagens recebidas, e se o emissor não receber nenhuma mensagem de *acknowledgement* durante um período fixo de tempo, *timeout*, assume que a mensagem perdeu-se e re-transmite-a de novo. O problema associado com esta metodologia é a duplicação de mensagens e o elevado tráfego na rede de comunicação de dados. Este método é importante nas falhas temporais do canal.

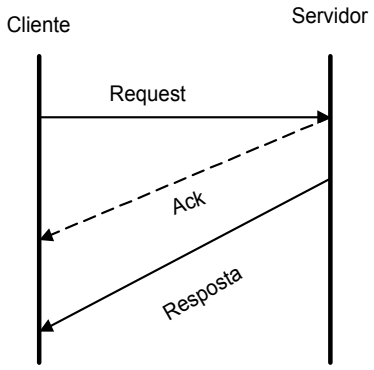


Figura 2.1. Invocação e ack

- Na replicação simétrica não existe um *master* designado para cada pedaço de dados. Uma replica pode ser usada como uma cópia actualizável em qualquer momento. Na replicação passiva os conflitos são determinados depois dos factos e são resolvidos no subscritor no secundário. Nestes casos são empregues algumas regras no secundário que ajudam a resolução de conflitos. Por exemplo, o uso de *timestamps* no secundário para identificar a cópia mais recente. Apesar destes regras serem aceitáveis, não permitem ao utilizador ver o estado global consistente de todo o sistema.

- *Mirroring*: um dos método muito usado para a implementação de tolerância a falhas é o *mirroring,* que é a forma mais simples da replicação. Neste tipo de implementação, quando a cópia original difere da cópia do *mirror,* é detectada a falha.

Muitos destes mecanismos de detecção de falhas são empregues em simultâneo, exceptuando alguns casos, devido à natureza das falhas e a sua especificidade de detecção. Os mecanismos mais utilizados nas bases de dados para detectar e recuperar a falha baseiam-se em transacções atómicas. Nos canais de comunicação as falhas são evitadas utilizando os mecanismos de auto-verificação (*bit* de paridade, *checksums* e os *timeouts*). Na replicação de dados a detecção de falhas e resolução de conflitos baseia-se na definição de prioridades sobre os dados (posse de dados) nomeadamente a técnica de *master/slave.* Existem ainda outros métodos de detecção e recuperação de falhas de acordo com as especificações de base de dados, incluindo os métodos dos ficheiros de *log* e outros definidos pelo utilizador.

Capitulo III

3. Técnicas de replicação

3.1. Replicação activa e passiva

As técnicas de replicação constituem um dos grandes focos de discussão das alternativas de sincronização dos dados distribuídos. Estas técnicas são determinantes na disponibilização de dados e na tolerância a falhas. As técnicas básicas dividem-se em dois grandes grupos: a replicação activa e a replicação passiva. A partir destas são vários os métodos de replicação que os fabricantes e fornecedores de *hardware* e *software* apresentam no mercado das tecnologias.

Na replicação activa, vários servidores funcionam em simultâneo para tolerar falhas e aumentar a disponibilidade total. É uma técnica considerada dispendiosa na implementação e gestão, mas que simultaneamente garante a melhor capacidade de resposta, sobretudo quando a disponibilidade é prioritária. É por isso é muito utilizada nos sistemas críticos onde uma falha do sistema pode resultar num desastre.

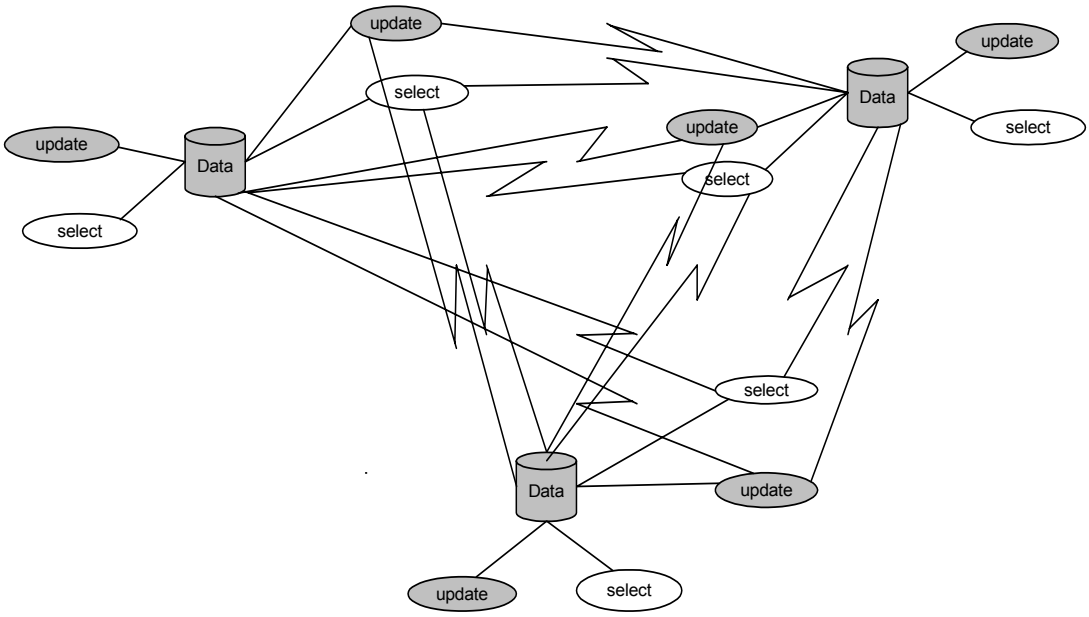


Figura 3.1. Replicação activa modelo *update anywhere*

Na replicação passiva, em cada instante um servidor é primário e executa todos os processos, pedido/resposta, enquanto um outro servidor fica em estado de espera, para a eventualidade de uma falha do primeiro. Em termos comparativos, a tolerância a falhas usando a replicação passiva tem duas vantagens principais sobre a replicação activa. Primeiro é mais simples durante a operação normal porque as mensagens são apenas enviadas a um servidor (o primário) e não a um grupo inteiro. O problema associado com a ordenação dessas mensagens desaparece. Segundo, na prática este requer poucas máquinas, porque em cada instante é sempre necessário apenas um servidor primário e um de *backup.*

Uma das desvantagens da replicação passiva é a sobrecarga nas comunicações que a técnica implica. Com efeito, a comunicação síncrona de verificação *online* requer muitas mensagens de comunicação em circulação usando o *multicast.* Uma outra limitação verifica-se quando o primário falha e há necessidade de recorrer ao segundo servidor, o que implica uma elevada latência na rede, perdendo-se a linearização e levando o cliente a ficar com uma consistência mais fraca [9,19].

Como exemplo de aplicação da replicação passiva refira-se o *SUN NETWORK INFORMATION SERVICE* (NIS). A SUN usa a replicação passiva para garantir a alta disponibilidade e maior desempenho, mas com fraca consistência. A fraca consistência é ainda aproveitada para gravar certo tipo de registos de administração de sistema [5]. Os dados replicados são actualizados no servidor *master* e propagados daqui para os *slaves* usando a comunicação de um para um (em vez da comunicação em grupo). Os clientes podem comunicar com cada *master* ou *slave* para obter informação.

3.2. Técnica de replicação de dados baseada em discos

A técnica de replicação designada por *stable storage* (dispositivos estáveis) agrupa-se no tipo de replicação passiva, sendo caracterizada pela duplicação dos dispositivos de armazenamento como forma de garantir um sistema de armazenamento estável. O sistema de discos consiste em conjunto de discos ordinários, sendo assumido que os mesmos se podem degradar independentemente. Nestes casos, cada bloco do disco tem uma cópia exacta num outro disco. Assim, as falhas inesperadas podem ocorrer, mas garante-se até um nível aceitável, que um

dos discos esteja operacional e não seja afectado. Normalmente, isto consegue-se implementando métodos de acessos diferentes para os dois discos. Para os discos convencionais, as operações básicas de leitura e escrita são executadas de um modo diferente. Suponhamos que temos dois discos designados por d1 e d2. Para uma operação de leitura, por exemplo, primeiro tenta ler a partir do disco d1, e se falhar então a leitura é feita a partir do disco d2. A escrita é efectuada nos dois discos, mas a escrita no disco d2 só começa depois de concluir com sucesso a escrita em d1. Esta é a maneira mais simples para evitar a possibilidade de dois discos ficarem danificados fisicamente ao mesmo tempo por causa de falha de *hardware*.

Existem acções de recuperação que restauram a consistência interna dos dados armazenados nos dois discos, depois de uma falha ter ocorrido. Estas acções de recuperação comparam o conteúdo dos dois discos bloco a bloco. Quando detectam que dois blocos correspondentes diferem, o bloco tendo dados incorrectos é regenerado a partir do bloco correspondente do outro disco. A identificação do bloco correcto depende do registo temporal associado ao bloco ou disco em falha [8].

Exemplo :A tecnologia RAID

Os métodos de replicação baseados em *Redundante Arrays of Inexpensive Disks* (*RAID*) baseiam-se na combinação de discos de menor capacidade e baratos para alcançar o desempenho e confiabilidade de um disco simples e cara. Existem várias implementações de RAID maior parte dos quais baseados no *hardware* como forma de tolerância a falhas. Uma dessas implementações é a implementada na TDM, o *mirroring* dos discos [4,25].

Para uma maior disponibilidade e tolerância a falhas no caso mais simples utiliza-se o RAID do nível 1 e 5 pois apresenta 100% da redundância tanto na escrita como na leitura. Uma falha do disco não requer uma acção de reconstrução, bastando uma cópia para o disco de substituição. A desvantagem que técnica implica é a elevada taxa de sobrecarga tornando-a ineficiente. A outra desvantagem relaciona-se com a sua implementação dado que nalgumas situações é feita utilizando o *software* e por conseguinte consome os recursos do servidor, o CPU. Nos tempos de elevada utilização do sistema o *throughput* degrada-se. Por esta razão uma implementação baseada no *hardware* tem vantagens.

Existem outras implementações de RAID que são diferentes do RAID de nível 1 e

orientados para objectivo específico.

3.3. Técnicas de *Snapshots* e *Triggers*

Snapshot são cópias de tabelas feitas automaticamente. Habitualmente estas cópias de tabelas estão numa outra base de dados [4].

Snapshots das tabelas permitem a distribuição assíncrona das modificações das tabelas, das vistas, ou porção das tabelas, com base num esquema pré-definido, por exemplo uma vez por dia. Um método comum de distribuição de *snapshots* é o uso dos ficheiros *logs* (*log flies)* nas bases de dados, que reduz a sobrecarga no sistema. A utilização de ficheiros *log* é uma boa fonte de registo das modificações nos dados originais. Pode-se criar um mecanismo que usa os ficheiros *log* para detectar modificações nos dados fontes e propagar mudanças para a base de dados alvo sem interferir nas operações normais do sistema fonte. Um servidor do SGBD pode conter os ficheiros *log,* mas noutros casos poderá ser melhor armazená-las num servidor externo independente, o que oferece um grau de segurança adicional. Esta solução é dispendiosa, mas dependendo da dimensão do sistema de informação e da importância dos dados, pode ser uma boa escolha. Uma fila de espera de processos é útil para enviar actualizações para os outros locais e manter uma consistência relaxada, porque quando ocorre uma falha de rede ou do local, a fila de espera pode garantir a actualização até ao restabelecimento das comunicações. A integridade, e a ordem de actualização devem ser mantida na altura da entrega.

Uma alternativa que permite ao utilizador construir as suas próprias aplicações de replicação usa os *triggers* das bases de dados. O método dos *triggers* faz com que utilizador seja responsável pela criação do seu próprio código, dentro de um *trigger,* que irá executar quando um evento específico ocorrer. Por exemplo, em *Oracle* um *trigger* pode ser usado para manter uma cópia de uma tabela no outro local. A figura 3.2 mostra uma listagem do código a inserir num *trigger* por forma a duplicar uma linha.

```
CREATE TRIGGER factura_after_ins_row
BEFORE INSERT ON Factura
BEGIN
```

```
    INSERT INTO Factura_Duplicate@Factura_Duplicate_Link
    VALUES (:new.Sno,
END;                                                    [3]
```

Figura 3.2. Código de um *trigger*

Este *trigger* é invocado sempre que uma linha é inserida na tabela Factura.

Apesar dos *triggers* serem mais flexíveis que os *snapshots*, tem muitas desvantagens que incluem a sobrecarga associada à sua gestão e execução; são executados em cada momento que uma linha é alterada na tabela mestre o que pode afectar negativamente o desempenho de um sistema. Além disso, se a tabela mestre é actualizada frequentemente, isso pode dar lugar a uma sobrecarga na rede e na aplicação. Pelo contrário, os *snapshots* podem ser programados ou executados manualmente. A activação de *triggers* não pode ser ainda invertida facilmente, no caso de um evento que aborte a operação ou que obrigue ao recuo (*rollback)*. Qualquer um dos métodos pode evitar o carregamento de grande número de replicação de transacções durante o pico de utilização do sistema. Os mecanismos de *refresh* em grupo são utilizadas para a actualização das tabelas relacionais replicadas.

A *Oracle* já implementa a arquitectura de cliente servidor e modelo *master/slave* usando a definição de ambiente replicado que consiste em: um grupo de replicação, a definição de *master site* e de *snapshots* (ver capítulo IV) [4].

3.4. Modelos de posse de dados na replicação

O conceito de posse de dados na replicação evita conflitos na actualização de dados e define prioridades de actualização, isto é, qual o local que tem privilégios para actualizar os dados. Os tipo principais da posse de dados são *master/slave*, *workflow*, e *update-anywhere* por vezes também designado por *peer-to-peer* ou replicação simétrica.

No modelo *master/slave* os dados são armazenados num *master* de modo assíncrono. O *master* disponibiliza os dados, o *slave* subscreve aos dados proprietários de *master* e só recebe as cópias no estado restrito de leitura. Somente um local pode actualizar a cópia *master* num conjunto particular de dados. São exemplos deste tipo de sistemas: Sistemas de Suporte a Decisão

(DSS), o *Distribuition and Dissemination of Centralized Information*, C*onsolidation of Remote Information* e Computação móvel [3].

A posse de dados *worflow* é um modelo de posse dinâmico, evitando conflitos de actualização em simultâneo. A *workflow* permite que dados replicados actualizados se possam mover de um local para o outro. Mas, em qualquer momento, só existe um local onde é possível actualizar dados específicos. O exemplo de posse de dados *workflow* é um sistema de processamento em cadeia, onde se segue um conjunto de passos bem definidos. Os sistemas centralizados permitem às aplicações acederem à actualização de dados numa base de dados integrada. Cada aplicação actualiza os dados numa determinada sequência, quando e somente quando o estado do ficheiro ou dados indicar que uma eventual operação anterior está completa. Este modelo permite que as aplicações possam ser distribuídas em vários locais onde e quando os dados são replicados e propagados para o local seguinte dentro de uma cadeia [3,19].

Os modelos *master/slave* e o *workfow* são restritivos porque em cada momento, somente um local pode actualizar os dados, tendo todos os outros locais apenas o acesso de leitura às replicas. Em certos ambientes este modelo é muito restritivo, sendo nesses casos, utilizado o modelo de replicação simétrica que cria o ambiente ponto a ponto. Este é caracterizado por múltiplos locais terem direitos iguais para actualização de dados replicados, dando mais autonomia local. Para garantir a consistência emprega-se o protocolo 2PC nas transacções. A execução concorrente das transacções é serializada pelos mecanismos dos *locks*. Quando se usa a replicação assíncrona com este modelo, as transacções de execução paralela podem interagir com a base de dados num estado inconsistente. Por isso, este mecanismo é frequentemente complementado com mecanismos de detecção e resolução de conflitos no dispositivo secundário. Os conflitos possíveis de acontecer são aqueles que ocorrem entre as tabelas e dentro das tabelas. O conflito entre tabelas ocorre quando dados relacionados são difundidos em dois ou mais locais. Frequentemente é resultado de um fraco desenho na fragmentação de dados ou da chave primária e/ou estrangeira nos locais remotos [8,19]. O conflito que ocorre dentro das tabelas, verifica-se quando os dados guardados de modo redundante em vários locais são considerados actualizáveis, ou seja, tem as mesmas prioridades.

Em ambos os casos, os conflitos são detectados depois dos factos e são resolvidos ou através de operações manuais, ou através de operações automáticas de compensação das

transacções, desfazendo as transacções anteriores. A compensação das transacções tem a inconveniente de violar a durabilidade das transacções, propriedade ACID das transacções.

Por isso, usar este modelo é o mesmo que ter o sistema centralizado sem mecanismo dos *locks*. Este modelo demonstra maior desempenho, mas a integridade é comprometida. Pode-se concluir que a replicação passiva com o modelo de posse de dados de replicação simétrica é vantajosa somente nos sistemas com baixa taxa de actualização.

3.5. Consistência e recuperação de falhas em bases de dados replicadas

Os mecanismos de controlo de consistência para garantir que uma operação executada num dado item seja reflectida nas cópias físicas desse item são alcançados com os critérios de serialização, sequencialização e causalidade de cada cópia (também designados por consistência linearizável, consistência sequencial e consistência causal). A serialização das cópias garante que um sistema replicado tem sempre a base de dados distribuída consistente, mesmo na presença de falhas do local, ou das comunicações. Uma execução diz-se serializável quando esta produz os mesmos *outputs* e tem o mesmo efeito na base de dados como alguma execução serial da mesma transacção [1,9]. Uma execução é serial se em qualquer par de transacções T1 e T2 as operações de cada transacção são executadas segundo a ordem da série S, consistente com o tempo na qual as invocações ocorreram e a sequência segue especificações de uma cópia correcta, isto é, se T1 contém as operações $T1=\{o_{10},o_{11},o_{12},\ldots,o_{1n}\}$ e $T2=\{o_{20},o_{21},o_{22},\ldots,o_{2n}\}$ a série $S=\{o_{20},o_{21},o_{10},o_{22},o_{11},o_{12},\ldots,0_{2n},o_{1n}\}$ representa a sequência das operações na ordem da série (execução) [9,24].

O critério da consistência sequencial considera a ordem em que os pedidos são atendidos, e não o tempo absoluto dos eventos. A consistência sequencial é serializável, mas o contrário não. A consistência sequencial é fraca comparada com a serializável e baseia-se na ordem total das invocações de todos os clientes. A serialização faz parte das propriedades ACID das transacções atómicas (Isolamento) e é aplicável nos sistemas estritamente precisos onde não é tolerado a inconsistência de dados. Este critério reflecte o caracter indivisível das transacções atómicas, se um grupo das transacções é executada de modo concorrente, o seu efeito é o mesmo como se elas tivessem sido executadas sequencialmente na mesma ordem [5]. A consistência

causal é um outro critério mais fraco comparada com os dois primeiros [24]. Muitas aplicações com dados replicadas requerem uma consistência forte (serializável ou sequencial) [1,24]. Na replicação, os critérios de consistência tem um papel especial na tomada de decisão da distribuição de dados porque um ambiente distribuído torna a consistência mais complexa, principalmente quando ocorrem falhas [5]. Uma avaliação dos custos da consistência "custos da complexidade" comparado com os benefícios da replicação de dados é fundamental, principalmente nos sistemas que implementam a técnica de replicação para elevar a disponibilidade porque a propagação das actualizações deve ser consistente e eficiente em todas as réplicas. Uma falha na avaliação de custos de consistência pode reflectir-se num fraco desempenho em todo o sistema.

Quanto à recuperação de falhas num sistema replicado existem já vários métodos, uns chamados de compensação da falha e outros recuperação da falha. O método de compensação da falha baseia-se na replicação de componentes que evidenciam as falhas. O método de recuperação da falha baseia-se na técnica das transacções atómicas (atomicidade) e complementado pelos métodos de controlo da concorrência, por exemplo o protocolo *Two-phase-locking* (2PL). Quando ocorre uma falha no sistema, as transacções que falharam são eliminadas na base de dados e o estado do sistema ou avança ou recua para um estado consistente. Este método implica a eliminação de certas etapas de processamento já efectuadas [5,7]. Num sistema distribuído o método comum utilizado na recuperação de falhas é baseado na atomicidade das transacções que considera dois locais, um local designado por coordenador (também participante) e um ou mais por participantes. Uma transacção em curso só termina o *commit* se todos os locais participantes estiverem de acordo. O processo do fim da transacção começa quando o coordenador envia a mensagem para todos os participantes a perguntar se pode fazer o *commit* da transacção (1ª fase), esta mensagem é registada no ficheiro *log* do coordenador. Os participantes respondem positivamente ou negativamente. Quando respondem positivamente, o coordenador envia a segunda mensagem a todos os participantes da tomada de decisão de *commit* da transacção, “mensagem de preparo” (2ª fase) e todos os participantes devem retornar uma mensagem *acknowledgement* (*ack*) ao coordenador que é registado no ficheiro *log* do local participante. Uma resposta negativa do participante acontece quando um ou mais participantes tiverem perdido algumas transacções. A maneira como o participante determina a resposta

correcta (positiva ou negativa) baseia-se no critério de controlo das falhas, o método do contador de falhas ou "número de encarnação" que é gravado em cada participante e transportado em cada mensagem do coordenador e/ou contido em cada transacção. Cada transacção (mensagem) que visita um local (participante) compara o seu contador de falha com o do participante. Se durante as duas visitas de uma transacção o "número de encarnação" diverge então ocorreu uma falha e a transacção aborta [5]. Neste caso, os dados ou avançam ou recuam para um estado consistente. Este método funciona bem quando a alta disponibilidade não é prioritário, mas tem um impacto nas operações activas quando há falha do coordenador porque interrompe todas as transacções em curso. Na replicação síncrona passiva este impacto é menor devido a demora na propagação das transacções.

3.6. Protocolos de propagação das actualizações na replicação

As técnicas de actualização de dados num sistema replicado baseiam-se nos protocolos de consenso (QC - *Quorum Consensus*) e na cópia primária (ROWA - *Read one Write All*). O protocolo de propagação ROWA executa a operação de leitura de um item de dados, numa cópia simples de dados e a operação de escrita é executada em todas as cópias existentes sobre este objecto. A propagação das actualizações com base no protocolo ROWA é muito utilizada porque garante a consistência e tem um bom desempenho nas operações de leitura embora seja fraco na escrita [1]. A outra limitação deste protocolo verifica-se quando ocorre uma falha nos canais de comunicação porque afecta as operações de escrita tornando a cópia indisponível.

Uma alternativa é o uso de protocolo Consenso *Quorum* (QC), que permite que as operações de escrita e leitura sejam executadas num subconjunto de locais activos, garantindo a sobreposição de w*rite Quorum* [1,5,6]. Suponhamos que existem n cópias de F. Para a sua leitura é necessário que no mínimo r cópias de F sejam consultadas (*read* quorum). Para a operação de escrita é necessário que no mínimo w cópias de F sejam escritas (*write* quorum) [8]. E deve-se observar a condição de maioria, $r+w>n$, a soma de operações de leituras e escrita de consenso devem ser maior que o número total das cópias. Esta condição garante que a intersecção não seja nula, que existe uma cópia comum de ficheiro que resulta em pelo menos uma actualização, num

par de operação *read/write* quorum e recebe o número da versão mais recente. A grande vantagem deste algoritmo é o facto de mascarar as falhas e não ser necessária alguma intervenção depois de ocorrer uma falha e na recuperação. O protocolo QC não requer protocolos complicados de recuperação, é flexível. Assumindo que as cópias são igualmente distribuídas pelos locais, um esquema de centralização completa consegue-se pela atribuição de votos em todas as cópias num local, e um esquema de distribuição (descentralização) total alcança-se pela atribuição de peso igual a cada cópia. Todavia, o protocolo QC é dispendioso para sistemas que raras vezes falham, por isso como alternativa adoptam-se algoritmos ROWA/QC, em sistemas considerados mistos que combinam os métodos ROWA para o período de funcionamento confiável e mudar para QC durante o tempo em que há falha ou do local ou dos canais de comunicação.

3.7. Comunicação em grupo

Contrariamente aos protocolos de comunicação em que há um emissor e um receptor de mensagens, nos protocolos de comunicação em grupo uma mensagem é enviada simultaneamente para vários receptores, que constituem um grupo. A técnica de comunicação em grupo é utilizada como forma de alcançar melhor tolerância a falhas e disponibilidade. Conforme a permanência de um processo num grupo de outros processos intervenientes pode se distinguir dois grupos de comunicação, estático e dinâmico. Um grupo é estático se os seus membros se mantém estável durante o tempo de vida do sistema. Num grupo estático, a falha de uma réplica ou processo não requer acções específicas de recuperação, o sistema mantém-se estável. Por exemplo, na replicação activa a falha de um processo é compensada pelo próprio sistema [9,24]. Um grupo é dinâmico se os membros do processo ou réplica podem falhar e/ou recuperar de uma falha e reintegrar os outros processos sobreviventes durante o tempo de vida do sistema. Por exemplo, na replicação passiva a falha do secundário é transparente ao sistema, mas a falha do primário exige outras medidas que incluem a designação de um novo primário [9,24].

O protocolo de comunicação em grupo de interesse para a replicação é o *multicast* [1,9]. O *multicast* consiste no envio de uma mensagem para todos os membros do grupo que estão em escuta. Os outros protocolos de comunicação em grupo são o *broadcasting* e o *unicasting*. Na

comunicação por *broadcast* mensagens contendo um endereço específico são distribuídas por todos os membros do grupo, enquanto que a comunicação *unicast* utiliza mensagens dirigidas a receptores específicos. Como para um grupo com *n* membros em vez de uma mensagem para todo o grupo (*multicasting*) seria necessário enviar *n* mensagens (*unicasting*), a replicação à custa de *unicast* só é aplicável para pequenos grupos. Todos os protocolos de *multicast* tem a inconveniente de serem ineficientes e podem congestionar a rede ou o canal devido ao elevado número de mensagens em circulação [1,5,9].

A comunicação em grupo (*multicast*) num grupo dinâmico obedece a critérios de ordem e acordo entre os membros do grupo. Para além disso, um grupo dinâmico tem um serviço de gestão das relações do grupo designado por *group membership service*. Este serviço é composto por um interfaces do grupo, detectores (notificação) de falhas e uma vista do grupo. Os detectores de falha num grupo dinâmico não são perfeitos porque baseiam-se na técnica das suspeitas para excluir um processo ou uma réplica que falhou [1,9,24].

Uma vista do grupo é constituído por um conjunto de estado dos membros do grupo. O estado dos membros do grupo (ou a vista) é representado por identificadores de processos ou replicas activas durante o tempo de vida do sistema e é ordenado pela sequência de integração. Uma nova vista é definida logo que um primário (servidor, réplica ou processo) tiver falhado ou recuperado e reintegrar os membros sobreviventes do grupo. A falha do primário e a sua substituição segue uma regra determinística. Cada vista é distribuída por todos os membros correctos do grupo e em caso de falha do primário as réplicas estudam a identidade do novo primário o que habitualmente acontece na replicação primário *backup*.

A recuperação das réplicas no grupo começa com o envio de uma mensagem a todos os membros sobreviventes “mensagem de reencarnação” com uma nova identificação da réplica ou processo.

Numa rede particionada uma falha é restrita a uma partição e torna-se mais fácil a sua resolução recorrendo-se a protocolos de consenso (QC). Quer dizer, a falha de um subgrupo notifica os outros membros do grupo sobreviventes da falha e são capazes de tomar a decisão de continuar com o processo ou suspender com base no critério de maioria [9,24].

As vistas possuem três propriedades ordem, atomicidade e não trivialidade, respectivamente.

Ordem :

- Se o processo *p* entrega a vista *v(g)* e depois entrega a vista *v'(g)*, então ~∃*p*≠*q* que entrega *v'(g)* antes de *v(g)*.

Integridade:

- Se o processo *p* distribui a vista *v(g)*⇒(então)*p*∈*v(g)*

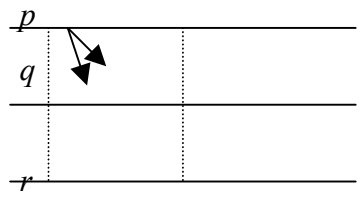
Não trivialidade:

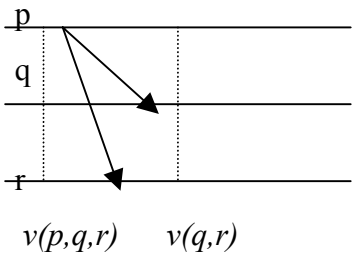
- Se o processo *q* junta-se no grupo e torna-se inalcançavel indefinidamente a partir de *p*≠*q*, então eventualmente *q* está sempre na vista que *p* distribui. De modo semelhante se a partição do grupo permanece particionada, então eventualmente a vista distribuída em cada partição vai excluir qualquer processo na outra partição [9].

Em todos os grupos, dinâmico e estático, a transferência do estado do grupo previne os outros membros da falha. O método utilizado para a passagem do estado é a distribuição das vistas por todos os membros [9,24].

A vista síncrona no grupo dinâmico oferece garantias de acordo (mensagens correctas entregam o mesmo conjunto de mensagens numa dada vista). Em outras palavras, se um processo entrega a mensagem *m* na vista *v(g)* e subsequentemente entrega a próxima vista *v'(g)*, então todos os processos que sobrevivem a falha a próxima vista *v'(g)* também entrega *m* a *v(g)*. As outras garantias evitam a repetição na distribuição de mensagem pela vista. Para ilustrar como funciona uma vista considere o seguinte exemplo: Suponhamos que *p* envia uma mensagem *m* e o processo *p* falha logo que envia *m*, assumindo *q* e *r* serem processos correctos [9].

Figura 3.3. vista de um grupo





1º caso: *p* falha antes de *m* alcançar qualquer processo *q* e *r* . Os processos *q* e *r* entregam a nova vista *v(q,r)* mas, nenhum dos processos vai receber a mensagem *m*.

2º caso: a mensagem *m* alcança pelo menos um dos processos sobreviventes da falha p. Então os processos *p* e *q* primeiro entregam a mensagem *m* e depois a vista *v(q,r)*

A técnica de vista síncrona nos grupos dinâmicos (replicação) é utilizado para a transferência do estado entre os membros do grupo designado por *view sincronous comunication*. A entrega da primeira vista contendo novos processos e antigos é iniciado pelo membro mais velho do grupo, enviando o seu estado utilizando a comunicação de um para um para novos membros do grupo e suspende a sua execução. O aspecto importante de sincronismo da vista na replicação é a introdução do conceito de paradigma de comunicação virtualmente síncrona originalmente desenvolvido nos sistemas ISIS [9,24].

3.8 Diferença entre a replicação activa e passiva

Uma replicação activa requer que as operações nas réplicas sejam deterministicas. Determinismo significa que o resultado de uma operação depende somente do estado inicial da réplica, e da sequência das operações precedentes pela réplica. Na replicação activa uma falha é transparente para o processo do cliente, o cliente nunca necessita de repetir o pedido. Na replicação passiva uma falha do *backup* é transparente para o cliente, mas o mesmo já não acontece com a falta do primário.

Capítulo IV

4. Estágio actual dos produtores de *software* e *hardware* de replicação

4.1. Introdução

As organizações para competir no mercado devem ser capazes de garantir a disponibilidade de dados do negócio. A disponibilidade num sistema informático está relacionada com outras dependências não funcionais que incluem a tolerância a falha e o desempenho. Actualmente, muitas organizações implementam tecnologias da *web* para oferecer os seus serviços ou produtos aumentando ainda mais a dependência entre os fornecedores de serviços e o consumidor. Por exemplo, um cliente da *web* espera ter o serviço disponível 24 horas durante 7 dias. A disponibilidade e a tolerância a falha neste tipo de sistema exige outro tipo de tratamento. No passado (até anos 80), os mecanismos utilizados para tolerância a falhas eram estritamente eficazes para a falha de componentes, em particular as falhas de *software*, falhas de *hardware* e erros humanos [25]. Todavia, as falhas catastróficas (desastres naturais e sabotagem) continuaram a constituir uma grande ameaça para muitas organizações. Esta classe de falhas é resolvida pelo método de redundância de dispositivos de armazenamento de dados em locais geograficamente distintos. A replicação de dados em zonas geograficamente distribuídas aumenta a disponibilidade de dados e é eficaz na tolerância de todo o tipo de falhas.

No mercado das tecnologias de bases de dados os fornecedores apresentam já alternativas de alta disponibilidade baseada na técnica de replicação. Esta técnica aumenta a tolerância a falha para todo o tipo de falhas, e aumenta a confiabilidade nos sistemas críticos pela compensação de falhas ou sua recuperação. Em seguida são apresentados três fornecedores de produtos de bases de dados que incluem a replicação nos seus produtos com o objectivo de apresentar o que existe no mercado em relação à replicação, que semelhanças existe entre os produtos e o que podem oferecer em termos de disponibilidade, tolerância a falha e desempenho. A escolha dos fornecedores IBM e *Sybase* é devida ao elevado grau de maturidade dos seus produtos e servirá

de padrão de comparação com os produtos da *Oracle,* o principal fornecedor dos produtos das bases de dados da empresa em estudo.

4.2. Estratégias de replicação na IBM

A IBM tem vários métodos de replicação que inclui tanto a replicação activa como a passiva. Replicação activa é implementada no DB2 *Universal Database* envolvendo uma base de dado global actualizada por várias réplicas de várias tabelas em diferentes locais de acesso. Este tipo de replicação é implementado por exemplo, nos bancos. Para detectar conflitos e recuperar falhas a IBM emprega a técnica de envio em simultâneo de pré- e pós-imagem de tabelas ou dados a actualizar e a identificação do utilizador que executa a actualização. A pré-imagem corresponde ao estado antes de actualização e pós-imagem depois. Esta metodologia de actualização torna fácil elaborar códigos que podem recuar o estado da base de dados desfazendo as actualizações efectuadas numa tabela.

A replicação da IBM é extensiva para casos considerados complexos onde a manipulação de dados é efectuada com base no SQL. O emprego de SQL permite utilizar a lógica procedimental no DB2 parte dos quais podem ser invocados, é o exemplo destes procedimentos, os *triggers* no DB2 *Universal Database* e funções definidas pelo utilizador. Os sistemas de bases de dados replicados são utilizados com bom desempenho para suportar a replicação entre os sistemas de apoio à decisão (DSS) e os sistemas OLAP.

A Encina é uma infra-estrutura de *software* de computação utilizada para desenvolver, executar e gerir negócios críticos na arquitectura cliente/servidor. *Encina* suporta a replicação síncrona e assíncrona da fila de espera das mensagens, designada por IBM/Transarc's Encina e o CICS (*Customer Information Control System*).

A *Encina* faz parte dos sistemas de informação escaláveis da primeira classe baseados na comunicação de ponto a ponto [19,22].

O serviço da fila de espera das transacções designado por *RECOVERABLE QUEUING SERVICE (RQS)* aumenta a tolerância a falha na rede e reduz o tempo de resposta empregando uma replicação assíncrona das transacções. Nos períodos de elevada utilização do sistema (picos), as transacções são guardadas na fila de espera, e quando baixar o volume das

transacções, são completadas. Este serviço pode ser empregue nas transacções complexas dividindo-as em mais pequenas e mais fáceis de gerir. O RQS actua como um *buffer* ajustando as transacções nos picos e nos vales de ocupação do sistema.

O RQS é ideal para aplicação *workflows* onde uma simples tarefa deve ser movida em cadeia de etapas de processamento. Antigamente este tipo de aplicação era executado em *off-line* com processamento em *batch* ou em *online* baseado no *hardware* de tolerância a falha principalmente quando a confiabilidade é questão principal [22].

Os maiores fornecedores dos serviços de telecomunicação utilizam RQS *Encina* para garantir a confiabilidade dos serviços de mensagens para os seus clientes. Os sistemas baseados no RQS são capazes de suportar milhões de chamadas em simultâneo. O RQS encaminha cada mensagem recebida e distribui pelos clientes com segurança.

Nos ambientes heterogéneos a *Encina* tem uma ferramenta de administração muito simples baseado no *Graphical User Interface* (GUI) a *Econsola*. A *Econsola* permite ao administrador da base de dados ver o sistema *Encina* distribuído de uma maneira interactiva. Os produtos como *Encina* e o *Lotus Note* na IBM apresentam elevado grau de tolerância a falhas baseada na replicação.

A *Encina* possui um *Structure File Server* (SFS) com *developer*, um *record-oriented file system* compatível com os sistemas de programação dos interfaces das aplicações do estilo ISAM/VSAM. O SFS está configurado para actualizações intensivas, com características de acesso *multi threaded*, transacções aninhadas e um endereço integrado seguro. Por isso, a IBM considera a *Encina* uma extensão de *Distributed Computing Enviroment* (DCE) porque a *Encina* acrescenta mais requisitos e mais serviços a DCE. Estes serviços incluem a gestão das falhas, capacidade transaccional e um conjunto de recursos para manter um grau de desempenho nas actualizações intensivas do sistema.

A *encina* corre em sistemas operativos de pequeno porte (*windows*) e de grande porte (u*nix*). A *Transarc* é uma corporação líder em soluções de arquitectura cliente/servidor e desenvolve o DCE para plataformas da *Sun-Microsystem*.

LOTUS NOTES a replicação no *Lotus Notes* permite duas réplicas trocarem as suas alterações/modificações de dados. O *Lotus Notes* utiliza o modelo de replicação simétrica, e diferencia uma cópia de dados de uma réplica, porque às réplicas são atribuídas uma identificação

(ID) no acto da criação da base de dados. A replicação é utilizada para colocar os documentos perto do utilizador (aumentar a disponibilidade), em particular para as organizações ou empresas com muitos escritórios ou departamentos geograficamente distribuídos. Os documentos podem ser actualizados de modo concorrente em diferentes locais, e o sistema suporta uma sincronização automática das réplicas, usando o controlo das versões para detectar as réplicas que foram alteradas e precisam de ser copiadas. O cliente também pode replicar uma porção de base de dados para o seu computador pessoal (computação móvel), permitindo uma disponibilidade total de dados desligando-se momentaneamente do servidor principal de base de dados. Um interface de programação das aplicações permite aos utilizadores de *Lotus* N*otes* básico, desenvolverem aplicações complexas distribuídas. O *Lotus* N*ote* tem ferramentas de ajuda ao desenvolvimento, o *LotusScript* (*dedicated scripting language*), os filtros que podem processar documentos e desenhar elementos que facilitam as tarefas na construção de interfaces gráficas do utilizador GUI [20].

Os sistemas de *Groupware*, em particular os *workflows,* são baseados no *Lotus* Notes devido a sua maior facilidade na manipulação de cadeia de documentos. A componente de Notes, uma base de dados de documentos hipertexto, pode armazenar os *links* dos dados provenientes de várias fontes incluindo as páginas da *web*, mensagens e imagens.

Os conflitos da replicação são resolvidos na base de comparação de datas. O *Lotus* N*ote* conserva a cadeia de cada documento que foi editado por dois ou mais utilizadores na última replicação. Para além deste método, o *Lotus* N*ote* guarda todas as versões dos documentos editados e assinala-os para serem revistos. Os conflitos possíveis de ocorrer são: os conflitos de salvar documentos e/ou conflitos de replicação. Os conflitos de salvar ocorrem quando dois ou mais utilizadores abrem e editam o mesmo campo do mesmo documento em simultâneo no mesmo servidor. O *Lotus Note* trata o primeiro documento guardado de *main one* e todos os outros de *response document*. Um utilizador manualmente adiciona as mudanças dentro do *response document* para o *main one* guardado.

4.3. Estratégias de replicação no *Oracle*

O *Oracle* considera três componentes de replicação:

- Replicação de objectos
- Replicação de grupo/grupo de replicação
- Replicação local

Replicação de objectos verifica-se quando um objecto (dados, vistas, *triggers*, índices ou sinónimos) da base de dados distribuída existe em vários servidores.

Na Replicação de grupo/grupo de replicação, um conjunto de objectos replicados forma um grupo de replicação, o que facilita a administração de objectos. Os objectos no grupo de replicação podem ser originados por vários esquemas de bases de dados, e os esquemas podem conter objectos que são membros de diferentes grupos de replicação. Cada objecto de replicação só pode ser membro de um único grupo [26].

Com a replicação local um grupo de replicação pode existir em vários locais. Um local de replicação pode ser do tipo mestre ou *snapshot*. Um local mestre mantém uma cópia de todos os objectos no grupo de replicação. Todos os locais mestres num ambiente de replicação *multimaster* comunicam-se directamente com um outro para propagar as modificações de dados e esquemas no grupo de replicação. Um local mestre designa-se também por grupo mestre. Cada grupo mestre tem somente um *master definition site* que serve como ponto de controlo na gestão do grupo de replicação e dos objectos no grupo.

Um local *snapshots* pode ser de leitura (*ready-only*) ou de actualização. Um *snapshot* de leitura baseia-se no modelo *master/slave,* e a replicação é num sentido. Os objectos replicados ou dados replicados são propagados periodicamente da tabela mestre para vários locais de acesso à bases de dados distribuídos. O *refresh* também é propagado da tabela mestre*,* de uma maneira transaccional consistente baseado no intervalo de tempo determinado, para os *snapshot* de leitura.

O *refresh* do *snapshot* pode ser incremental ou completo. Um *refresh* incremental ou *fast refresh,* é utilizado para realizar o *refresh* nos *snapshots* simples. Os *snapshot log* nas tabelas mestres são empregues para registar as modificações que ocorrem nas tabelas mestres e actualizadas através de *triggers*. As mudanças captadas nesse *log* são usadas para realizar um *fast*

refresh de um *snapshot* simples. Por outras palavras, somente as linhas mudadas são usadas para actualizar as réplicas secundárias. Pelo contrário um *snapshot* complexo ou um *snapshot* simples sem o *snapshot log* no mestre deve ser feito o *refresh* usando uma regeneração completa a partir da tabela mestre.

Um *Snapshot* actualizável é aquele que permite a inserção, remoção e actualização dos dados ou linhas de uma tabela mestre alvo. Quando o *snapshot* actualizável é activado são adicionados dois objectos no acto da criação do *snapshot* local. Uma tabela para guardar a identificação da linha (ROWID) e o *time stamp* da linha actualizada dentro do *snapshot,* e um *AFTER ROW triggers* no *snapshot* da tabela base. Este serve para inserir o ROWID e o *time stamp* da linha actualizada e/ou apagada dentro do *snapshot* actualizável. Quando o *snapshot* é criado como um objecto replicado, o *Oracle* cria um *triggers* adicional associado ao programa no *snapshot* da tabela base. Este *triggers* é usado para chamar o procedimento gerado no local mestre para fazer a aplicação das mudanças. Se o utilizador selecciona uma replicação síncrona, então o programa de *triggers* faz RPC. Se o utilizador selecciona a replicação passiva, então o programa de *triggers* insere as transacções em deferido dentro da fila de espera no local do *snapshot*. No *snapshot* de leitura o *Oracle* cria uma vista de leitura das tabelas bases subjacentes, mas que o *snapshot* de actualização, esta vista é redigível.

O *oracle* tem duas formas de implementar a replicação dos dados : a replicação básica implementado utilizando a declaração *create snapshot* ou *create materialized view*. A replicação básica só pode replicar os dados (não os procedimentos e os índices), a replicação é somente num sentido e os *snapshots* são do tipo *read-only*. A forma avançada suporta várias configurações de replicação de *snapshot* actualizável *multi-master* e replicação simétrica. É muito mais difícil comparada com a configuração básica, mas tem a vantagem de replicar mais dados e mais objectos da base de dados.

Master site

SQL>*create MATERIALIZED VIEW log on table* nome_tabela;

Snapshot site

SQL> *create MATERIALIZED VIEW* nome_tabela

Refresh fast with primary key

Start with sysdate

*Next sysdate+1/(24*60)*

*As (select * from* name_tabela);

Figura 4.1. Código de um *snapshot*

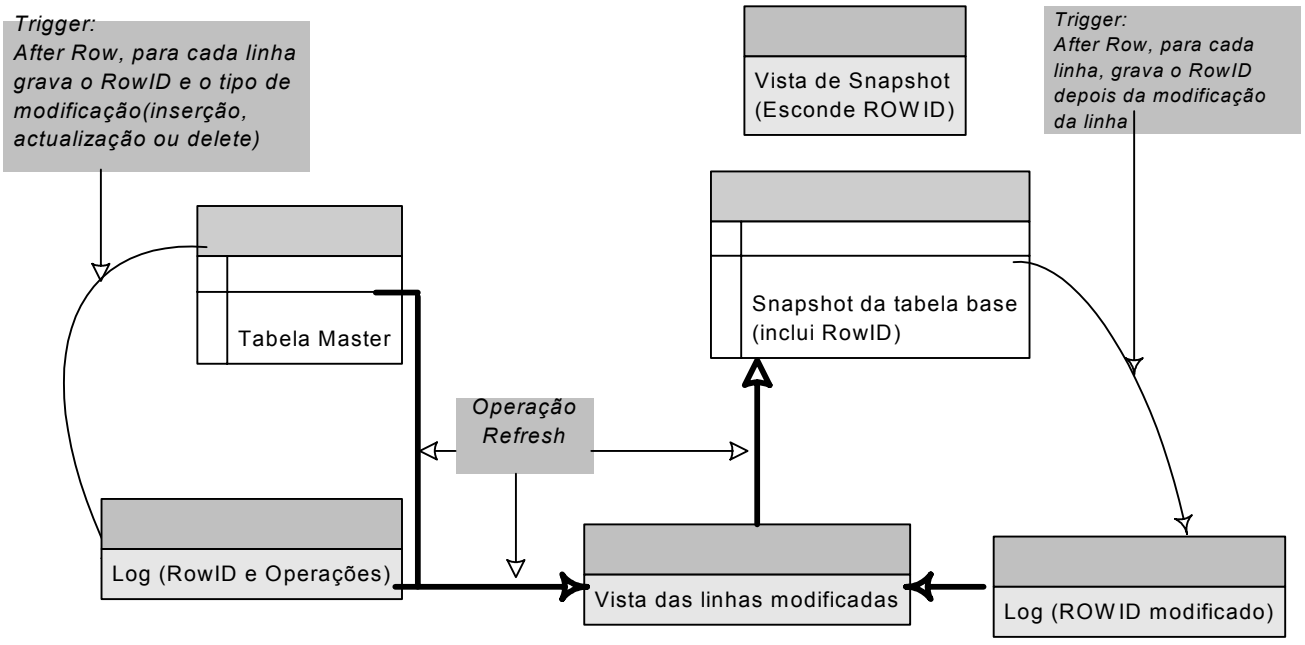


Figura 4.2.[1] *Snapshot* actualizável

2 Figura retirada de *Strategies and techniques for using Oracle 7 Replication, Oracle corporation*

O *Oracle* implementa a replicação simétrica no modelo de replicação simétrica (*update-anywhere)* podendo ser síncrona ou assíncrona. A arquitectura da replicação síncrona simétrica baseia-se nos *triggers*. Quando os dados são modificados o *Data Manipulation Language* (DML), (declarações INSERT, UPDATE ou DELETE), podem ser propagados usando ou o nível das linhas ou a replicação procedimental. Quando se usa a replicação a nível das linhas o *Oracle* gera os *triggers* e *stored procedure* para replicar cada transacção. Quando é executada uma instrução DML no local primário o *software* da replicação dispara os *triggers* resultantes da chamada do procedimento no programa gerado em cada réplica. A replicação ao nível da linha garante que todas as mudanças para uma tabela, se são a partir de instrução de DML ou *stored procedure*, sejam propagadas para todas as réplicas secundárias.

A replicação procedimental, permite que somente a chamada ao *stored procedure* faz a aplicação das actualizações para uma tabela designada em cada réplica secundária (replica somente o *stored procedure* utilizado pela aplicação para replicar as tabelas).

Os produtos e ferramentas da replicação simétrica em *Oracle* detectam a ocorrência de conflitos de actualização e automaticamente invocam uma aplicação específica de rotina de resolução de conflitos para dados replicados e colocar os dados num estado consistente. Quando os dados são propagados por um *pushing* a fila de espera das transacções deferidas chama o procedimento gerado no programa no dispositivo local de recepção. O argumento do procedimento inclui o valor antigo e o novo de cada linha e coluna. As linhas inseridas não têm valor antigo e as removidas não têm valores novos. O procedimento compara os dois valores (antigo e novo) de cada linha com o valor corrente no receptor. Os conflitos são detectados quando ocorre quaisquer diferenças dos dois valores nas colunas ou linhas em comparação. Todos os conflitos não resolvidos são introduzidos na tabela de erros do receptor.

Para a replicação da linha, o utilizador pode designar uma ou mais rotinas de resolução de conflitos que são aplicadas em ordem à prioridade. As rotinas podem ser suportadas pelo *Oracle* ou desenvolvidas manualmente. Para a replicação procedimental, o utilizador deve suportar métodos de resolução de conflitos como parte de procedimento de replicação.

Os *triggers* na tabela fonte fazem cópia das mudanças para dados com propósitos de replicação, guardando as mudanças necessárias numa fila de espera. O catalogo de replicação é um conjunto distinto de tabela de dicionário de dados e vistas que mantém informação

administrativa acerca de replicação de objectos e replicação de grupo no local. A limitação dos produtos da *oracle* é não permitir uma replicação de dados num ambiente heterogéneo, com outras bases de dados que não sejam do *oracle*, por exemplo o DB2, *informix*, *syabase* e *sql server*.

4.4.Estratégias de replicação no *Syabase*

Em resposta às exigências do mercado a *Sybase* apresenta uma gama de produtos com estratégia específica de disponibilidade e tolerância a falha. Neste caso para a alta disponibilidade, apresenta o *Adaptive Server® Enterprise* (ASE) 12.5 e os *Clusters*; para minimizar o tempo de falha (*downtimes*) e recuperação nas falhas catastróficas, emprega as estratégias de *OpenSwitch* e servidor de replicação (RS); para uma disponibilidade contínua de bases de dados e serviços utiliza o *Backup ASE* (em *online* ou *off-line*) e *File Transfer*.

- O ASE 12.5 é uma ferramenta de manutenção em online podendo ser utilizado na execução simultânea de várias tarefas incluindo o *backup*, reorganização das tabelas, re-indexação e isolamento das falhas. Para isso, o ASE 12.5 tem uma configuração dinâmica e a ferramenta *quiesce*. A configuração online da *Sybase*, mudou drasticamente as anteriores filosofias de configuração estáticas na qual os parâmetros configurados só tinham efeitos depois de *restart* da base de dados.
- *Quiesce database* - os comandos ASE *quiesce database* suspendem as actividades de inserção e actualização para uma base de dados específica. Uma base de dados *quiesce* (base de dados com o conteúdo congelado), permite ao DBA correr *backup* de dados em *online*. Os utilizadores podem continuar a aceder à base de dados, no estado de leitura (o *quiesce database* deixa a base de dados no estado leitura*)*. O ASE guarda um ficheiro *log* das actualizações que serão escritas na base de dados quando estiver liberto. Para além deste produto o ASE 12.5 suporta um sistema de ficheiros os *snapshots*. Um DBA pode tomar um *snapshot* várias vezes por dia, fazendo o *backup online* de toda a base de dados em pouco tempo. Esta facilidade é vantajosa porque os dados do *backup* estarão mais próximo do estado corrente da base de dados antes da falha. A gestão de *snapshot* é feita com ajuda de *data sotorage* do ASE incluindo

Network Application Filer, *EMC Symmetrix*, *Time Finder* e *Veritas Database Edition* para *Sybase*.

- *Clusters* – é uma técnica baseada em *hardware* ou servidores de base de dados preferencialmente para alcançar alta disponibilidade e reduzir o tempo de recuperação depois da falha. Um *cluster* tipicamente tem dois ou mais locais com os servidores de bases de dados activos. A inconveniência dos *clusters* é de os protocolos utilizados entre dois locais (servidores) serem habitualmente via SCSI ou outro protocolo que não é adequado para comunicações na WAN ou Internet. Por esta razão, os *clusters* devem estar próximos um do outro [21]. Esta situação faz com que os *clusters* sejam complementados com outras estratégias de disponibilidade de bases de dados que incluem um *datacenter* geograficamente distribuído.

- A *Sybase* normalmente utiliza dois *clusters* de *hardware*. Em cada local corre um ASE, configurado como servidor companheiro. Quando um servidor falha o outro automaticamente assume as operações da base de dados e as conexões de todos os utilizadores.

- *Hot-standby* activo/activo permite configurar dois servidores ASE como companheiros. Esta configuração permite uma configuração de companheiros assimétricos (*master/slave*) ou simétrico (activo/activo). A última configuração (activo/activo) em cada local é designado um *hot-standby* e oferece um *fast failover* com o mínimo de tempo de recuperação, exceptuando as falhas catastróficas. A *Sybase* utiliza outra técnica, a *seamless failover e failback,* que emite mensagens indicando a ocorrência de falha.

- A *Sybase* tem certificados de compatibilidade com a *Compaq, TruCluster, Hewlet-Packard Service Guard, IBM HACMP, Microsoft Windows NT MSCS* e a *Sun Microsystem Sun Cluster, Veritas cluster Server*, e a *Veritas Database Edition* para *Sybase*.

- O Servidor de Replicação (RS) é uma aplicação da *Sybase open server application* que oferece um serviço de distribuição e coordenação de todas as actividades de replicação. O RS faz a troca de dados entre o primário e o *standby database* e monitora continuamente os *logs* das transacções na base de dados primário e depois, faz o *push* para o *standby database server* através de um interface optimizado para a implementação específica de *warm standby*.

- A *Replication Agent Thread* ou *Log Transfer Manager* (LTM) é um programa embebido no motor da base de dados ASE, onde podem elevar o *throughput* no processo de extracção e entrega das transacções em curso. O RS é um sistema aberto e pode interagir com

várias bases de dados. Uma das facilidades de RS é a capacidade de replicar as alterações de *Data Definition Language* (DDL), e os *stored procedure* que podem fazer o *mirror* de toda a base de dados. Um *switch* muda a direcção da replicação. Depois da falha, o *switch* pode retornar ao primário e garantir o *pull* das transacções acumuladas no *standby database,* para a sincronização do primário. O *warm standby* não requer a definição nem a subscrição dos objectos de replicação, o que em geral simplifica a implementação e a gestão do ambiente *warm standby*. O *warm standby* é construído para arquitectura de rede de comunicação largamente distribuída e o RS aplica a função de integridade das transacções para a comunicação WAN, por essa razão o servidor de *standby database* é localizado num ponto geograficamente distinto da base de dados primária.

- *OpenSwitch* fornece as falhas automáticas ao utilizador. O *OpenSwitch* monitora a disponibilidade do servidor, quando o servidor falha, o *OpenSwitch* automaticamente transfere o utilizador (cliente) para o outro servidor. Para além disso, o O*penSwitch* faz o *restore* das conexões dos clientes para o servidor original após a recuperação da falha. Deste modo a combinação de RS e *OpenSwitch* fornece as falhas automáticas e as facilidades de *failback* que são necessárias para uma disponibilidade contínua.
- *Cold standby* implementa o *backup* periódico e *dump* das transacções do servidor da base de dados primário fazendo o *restore* no *standby*. Para o suporte deste processo o ASE 12.5 faz o *compress* dos dados acima dos 80% durante o processo de *backup*, o que reduz as necessidades do espaço de armazenamento e largura de banda durante o backup.

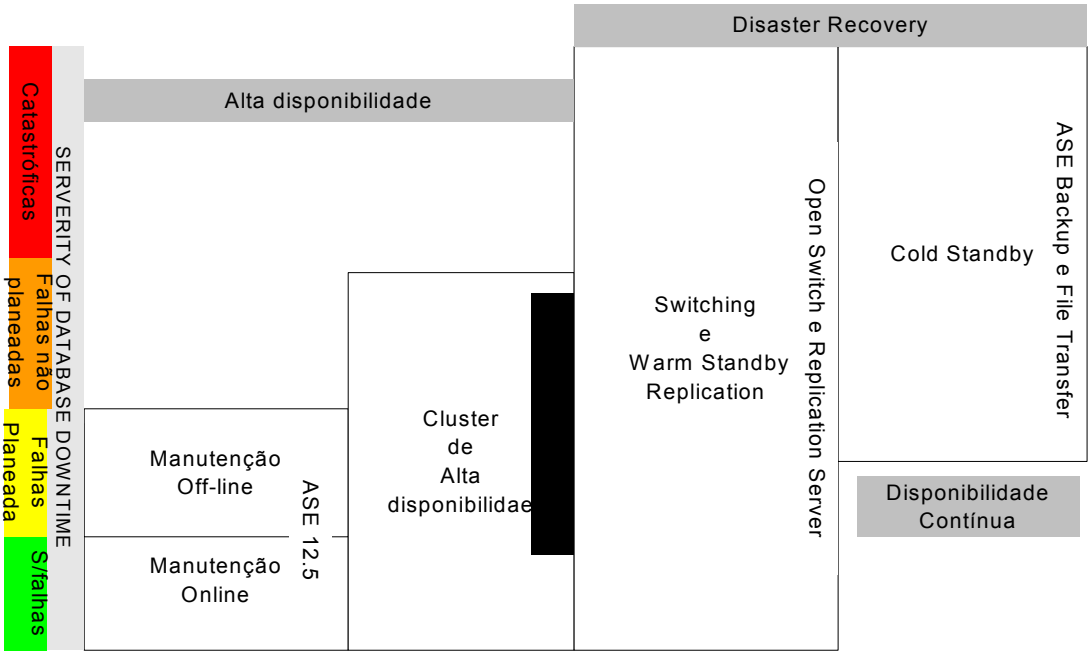


Figura 4.3. [2] Serviço de alta disponibilidade da *Sybase*

[2] Figura retirada do relatório *Strategies And Sybase® Solution for Database Availability*

Capítulo V

5. Caso de estudo

5.1. Apresentação da empresa e dos Sistemas de Informação

A TDM é uma empresa cuja missão é pôr as pessoas a comunicar e a desenvolver Moçambique através da venda de serviços de telefone fixo, como seu produto chave. A TDM é administrada e gerida por uma equipa dinâmica de gestores que acompanham as mudanças tecnológicas que ocorrem na indústria global das telecomunicações. Graças a esforços da coordenação de equipa de gestores com os seus 2308 colaboradores que compõem os seus recursos humanos, a TDM tem a centrais da sua rede 100% digitalizadas. A TDM detém o monopólio da telefonia fixa com cerca de 127902 clientes [28].

5.2. Sistemas de Informação

Os Sistemas de Informação (SI) na TDM tem desempenhado um papel fundamental na flexibilização dos processos da empresa e manuseamento de grandes quantidades de Informação em todos os seus departamentos. Os SI da empresa são abertos mas baseados na política de administração centralizada. A TDM tem duas bases de dados de importância vital nos processos do negócio da empresa (venda e entrega de serviços). A primeira é a base de dados dos clientes de telefone fixo (aplicação GIRAFE). E a segunda é a base de dados de logística (ORACLE FINANCIAL). Todos os processos de relacionamento com os clientes do serviço telefónico fixo, e com os fornecedores da empresa dependem das aplicações GIRAFE e ORACLE FINANCIAL.

O serviço de Internet e Intranet, centralizado em Maputo mas acedido de todo o país, cresceu, e como a TDM é um *provider* de Internet para os ASP em Moçambique, a necessidade de manter este serviço sempre disponível também aumentou. Para além disso a empresa vende o serviço de Internet a clientes singulares (Internet cafés) em quase todo o país. Nestas condições, a rede de comunicação de dados desempenha um papel fundamental.

5.3. Os desafios da empresa

Os grandes desafios que se colocam à TDM são a melhoria do desempenho, o aumento da disponibilidade e a tolerância à falha (incluindo segurança de dados) das aplicações e bases de dados. Os mecanismos de tolerância a falhas e segurança de dados que a empresa implementa não satisfazem as exigências actuais, sobretudo no que diz respeito a falhas, fiabilidade e desempenho. Por exemplo, a disponibilidade e o desempenho é afectado pela cadeia de *hardware* de acesso de dados centralizados criando diferenciação na oferta dos serviços aos clientes e fornecedores. Isto é, nalguns locais o sistema apresenta bons indicadores de desempenho, fiabilidade e disponibilidade, mas noutros locais o serviço é lento, com menor confiabilidade e disponibilidade.

Os planos de interrupção da exploração da base de dados (salvaguardas, facturação, manutenção dos servidores ou *upgrade* de *hardware/software*) também contribuem para a baixa disponibilidade das aplicações. Consequentemente a disponibilidade diminui e quando há falhas esta situação pode ser mais complicada.

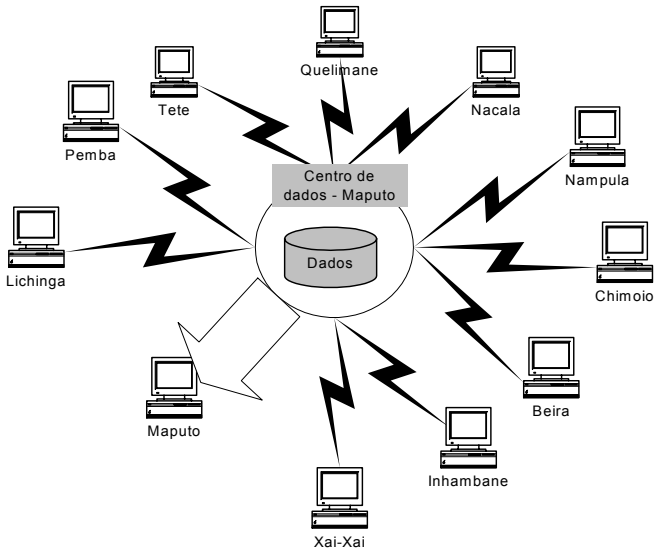


Figura 5.1. WAN e LAN da TDM

No que se refere à rede de dados pode-se identificar dois desafios, o primeiro que corresponde às falhas de facto e o segundo que corresponde a mecanismos da detecção de falhas e métodos de recuperação.

A figura 5.1 mostra o estado actual dos Sistemas de Informação da TDM. É uma *World Area Network* (WAN) de tipo estrela com várias LANs integradas que se comunicam com o centro de dados em Maputo.

5.4. Descrição dos problemas principais

As soluções que a empresa implementa para tolerância a falha (segurança de dados) e disponibilidade apresentam os problemas seguintes:

Os servidores de dados estão centralizado em Maputo, num único ponto. No entanto, uma redundância da base de dados requer no mínimo dois dispositivos separados de armazenamento de dados para a protecção de falhas.

Nos pontos de acesso remoto, todas as aplicações apresentam uma baixa disponibilidade devido a dependência da rede e dos servidores.

Em relação às salvaguardas (*backups)* o tempo necessário para a sua realização e recuperação é relativamente longo, o que sugere novos métodos menos dispendiosos em tempo.

Falta padronização dos processos de entrada e saída de dados (leitura de bandas magnéticas de chamadas telefónicas e impressão de facturas de consumo) devido a inflexibilidade da centralização dos Sistemas de Informação.

Por exemplo, a leitura de contadores e das chamadas telefónicas feitas pelos clientes em cada central telefónica do telefone fixo tem um tratamento diferente. Nas áreas de acesso remoto, as chamadas são gravadas nas *disketes* ou em discos ópticos ou ainda em bandas magnéticas. A nível da empresa existem três centros onde os dados podem ser gravados (Maputo, Beira e Nampula) e copiados para a Base de dados. Apenas Maputo é que pode ler as bandas magnéticas sendo que os outros locais ou enviam para Maputo ou fazem *ftp* para transferir o ficheiro para o servidor da base de dados.

Outro exemplo é dado pela impressão das facturas dos clientes nos locais de acesso remoto, que se faz por *ftp* para o servidor da aplicação onde depois são imprimidas. Este processo

é antecedido do envio de um *e-mail* em Maputo para o utilizador no local remoto, informando que os ficheiros da factura já estão disponíveis.

Os procedimentos acima apresentados mostram um sistema passível de falhas que pode dar origem a erros humanos. Um erro humano pode resultar em custos elevados para a empresa.

O serviço de Internet, Intranet e de correio electrónico (*e-mail*) é disponibilizado a todos os trabalhadores da empresa e pode servir de um meio alternativo de comunicação e divulgação da Informação para todos os trabalhadores, nomeadamente ordens de serviços e correspondência. No entanto, a utilização destes serviços torna-se dispendiosa para a empresa devido aos vários contactos que um cliente deve estabelecer com o servidor (primário) em Maputo para consultar ou enviar *e-mail*, principalmente para as áreas de acesso remoto. Estes problemas são agravados pela falta de um *proxy* local, falhas de hardware de comunicação e elevado tráfego na rede de dados tornando o sistema lento.

5.5. Levantamento e caracterização do SI da empresa

A tabela seguinte mostra a distribuição dos clientes, utilizadores, transacções efectuadas por local durante o primeiro semestre de 2002 na aplicação Girafe.

Zona	Local	Clientes	%	*Users*	%	Nº PCs	Total/Transac	%	Leituras/mês	%	Avarias/mês
Sul	Maputo	42,257	63%	286	51%	652	14,910.0	69%	70129.3	42%	3794.3
	Xai-xai	1,823	3%	22	4%	13	563.8	3%	5216.7	3%	131.2
	Inhambane	1,423	2%	21	4%	23	687.3	3%	4587	3%	114.7
Total/zona		45,503	67%	329	59%		16,161.1	75%	79933	48%	4040.2
Centro	Beira	7,065	10%	55	10%	48	1,467.3	7%	35683.0	21%	250.5
	Chimoio	2,380	4%	28	5%	19	686.8	3%	3216.7	2%	50.3
	Tete	2,087	3%	30	5%	19	834.8	4%	6328.0	4%	96.5
	Quelimane	2,799	4%	27	5%	19	757.0	4%	8716.7	5%	255
Total/zona		14,331	21%	140	25%	105	2,988.8	14%	53944.3	32%	652.3
Norte	Nampula	3,872	6%	30	5%	20	824	4%	13616.7	8%	138.3
	Nacala	1,334	2%	21	4%	18	659	3%	11800.0	7%	103.5
	Pemba	1,528	2%	21	4%	13	508	2%	4900.0	3%	32.7
	Lichinga	1,028	2%	17	3%	22	471	2%	2950.0	2%	42.2
Total/zona		7,762	11%	89	16%	73	2,461	11%	33266.7	20%	316.7
Total		67,596	100%	558	100%	178	21,611	100%	167,144	100%	5,009

Tabela 5.1. Distribuição dos clientes, utilizadores e transacções na aplicação Girafe

5.6 Caracterização da tolerância a falha e segurança de dados, disponibilidade e desempenho dos SI da empresa

Existem na empresa quatro mecanismos de tolerância a falha e segurança de dados:

Backup diário e mensal nos tapes

Um secundário

Mirror dos discos

UPS (*Units Power Supply*)

Os *backups:* há dois tipos de *backup* em *offline* (nas bandas magnéticas) um diário e o outro mensal (comando *tar* do *Unix*). O *backup* diário é feito nas noites e contém dados com BD em cima. O *backup* mensal contém dados e programas e é realizado no final de cada mês com BD em baixo. Apesar deste processo ser contínuo e garantir a segurança de dados em caso de falha, as bandas magnéticas estão armazenados no mesmo local de dados primários. Na eventualidade de uma falha que obriga a recuperação de dados pode tornar impossível, ou pelo menos será um processo relativamente longo e ineficiente para a disponibilidade de serviços em menos tempo.

Um servidor secundário (*standby database*) – é um servidor de espera que transfere os dados do servidor da base de dados do cliente (aplicação Girafe) para o de espera. É uma medida que guarda os dados muito próximo do estado corrente da base de dados. Todavia, a tolerância a falha utilizando esta metodologia falha porque a transferência de dados do servidor de produção para o de espera é feita com base num código elaborado em *shell* e não uma replicação baseada nos produtos da *oracle*. E ainda o método de transferência de dados consome o espaço no disco do servidor primário porque os ficheiros primeiro são arquivados e depois transferidos para o *standby database*. Além disso, o secundário está no mesmo local onde está o primário. As consequência são as falhas frequentes da *standby database* e a falta do isolamento na eventualidade de falta de serviços de energia ou telecomunicações.

A aplicação OF depende de um único servidor e com um único mecanismo de segurança de dados baseado nas bandas magnéticas.

Mirror dos dados : os discos dos servidores das bases de dados de cliente e da aplicação *Oracle Financial* tem uma configuração *mirror*. Os discos são de tipo SCSI e FC, é um sistema de escrita dual com um bom desempenho.

UPS e grupo gerador : a TDM utiliza dois mecanismos de tolerância a falta de energia: uma UPS com uma autonomia de duas horas de tempo e um grupo gerador com uma capacidade ainda superior a da UPS.

A solução de disponibilidade de bases de dados que a empresa construiu satisfaz os requisitos do negócio, quando não ocorre nenhuma falha. Como a disponibilidade depende da infra-estrutura de comunicação, do estado dos servidores e das bases de dados, o grande desafio que tem afectado a empresa é aumentar a disponibilidade e o desempenho das aplicações principalmente nos pontos de acesso remoto. As falhas mais frequentes verificam-se na infra-estrutura das comunicações e as interrupções planificadas para manutenção do sistema ou para processamento de facturas (e/ou manutenção).

Algumas iniciativas da Direcção dos Sistemas de Informação em coordenação com a gestão do topo da empresa já contemplam outros métodos de pagamento que não interagem directamente com a base de dados como forma de diminuir a procura dos serviços da base de dados, em particular nos finais de cobrança de facturas de telefone (aplicação GIRAFE). Este processo apresenta falhas porque a actualização do pagamento na base de dados é feita com base na transferência de ficheiros.

Numa perspectiva de *hardware* a aplicação Girafe apresenta uma arquitectura *3-tier* e compreende um servidor de base de dados, um servidor da aplicação e o cliente, figura 5.2.

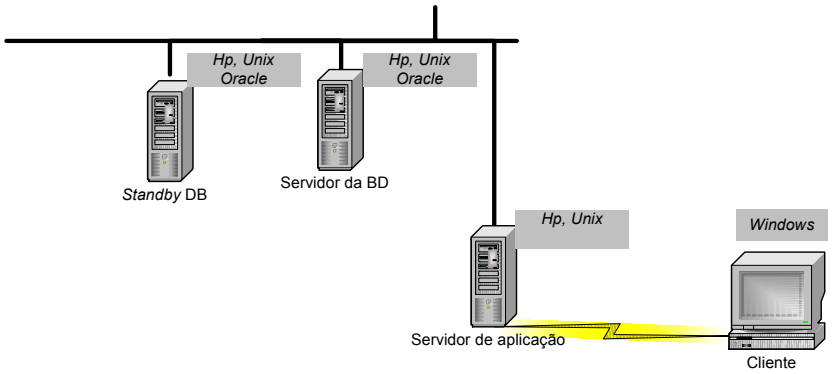


Figura 5.2. Arquitectura da aplicação Girafe

A aplicação OF apresenta uma arquitectura *2-tier*, conforme ilustrado na figura 5.3.

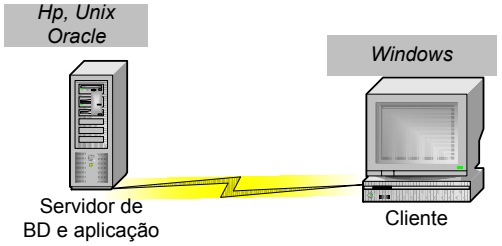


Figura 5.3. Arquitectura da aplicação OF

Os canais de comunicação tem uma capacidade que varia entre 64 a 128 K nos pontos de saída. A aplicação Girafe mostra uma melhor flexibilidade e resultando num bom desempenho a nível dos utilizadores locais (Maputo) em comparação com a OF. A aplicação OF apresenta problemas de desempenho (elevado tempo de resposta), disponibilidade e confiabilidade uma vez que todos os processos correm na mesma máquina.

5.7. Bases de dados

Existem duas bases de dados principais que são acedidas de todo o país (acesso remoto):

- Base de dados dos clientes
- Base de dados da *oracle* financial

Base de dados dos clientes

A base de dados dos clientes de telefone fixo contém cerca de 156922 registos em 1170 tabelas. A base de dados como já foi referido, é independente da aplicação, possui arquitectura *3-tier* e corre num servidor a parte. A estrutura das tabelas de dados é relacional. O método de acesso à dados baseia-se nos índices. Algumas transacções que tem lugar na BD são: registo de uma requisição do pedido de uma nova instalação de telefone; assinatura dos contratos de telefone; leitura e inserção das chamadas telefónicas dos clientes; pagamento do consumo dos clientes; reclamação dos clientes; participação das avarias dos telefones; reparação das avarias dos telefones; informação de assistência aos clientes. A base de dados é centralizada e é acedida

em todo o país, sendo portanto o serviço vulnerável a falha das comunicações. Os casos mais críticos são por exemplo nas avarias de telefones. A avaria dum telefone é participada via telefónica para o 170 em cada local (incluindo acesso remoto), regista-se na base de dados em tempo real (*online*), depois se emite uma ordem de trabalho para as brigadas de reparação (fornecedor de serviços, TELEVISA), só mais tarde é que se repara o telefone. Quando há falha do sistema ou das comunicações, não se pode reparar o telefone.

Problemas semelhantes verificam-se na informação de apoio ao cliente (telefone 130), reclamação dos clientes, gestão das impressões cuja gestão não requer um controlo centralizado.

Base de dados de *oracle* financial

A base de dados da *oracle* financial é constituído por 4500 registos e 9717 tabelas. É uma base de dados que suporta a aplicação no modo gráfico. A base de dados e a aplicação correm num único servidor, arquitectura *2-tier*, e os acessos são efectuados via *web*. A maior parte dos processos da base de dado OF são *batch processing*. Algumas transacções são: requisição do material para a TDM; introdução das facturas dos fornecedores, aprovação das facturas dos fornecedores, pagamento das facturas e impressão dos cheques.

Nos pontos de acesso remoto, a aplicação regista elevado tempo de resposta. A disponibilidade e o desempenho é ainda mais baixo comparado com a outra aplicação devido a própria arquitectura.

As implicações da centralização de dados nesta aplicação são semelhantes às da aplicação Girafe. Por exemplo, um fornecedor local, fornece um produto ou um material requisitado pela TDM no Centro ou Norte de Moçambique, mas o seu pagamento está condicionado a disponibilidade da base de dados em Maputo. Esta dependência também não é transparente para o fornecedor.

5.8. Rede de comunicação de dados

A rede de comunicação de dados da TDM é do tipo estrela e compreende os servidores da base de dados e aplicação, um *switch* e um *router* centrais. No lado do cliente, compreende um

router, um *hub* e um interface do utilizador. Figura 5.4 mostra uma vista geral de cadeia de *hardware* de comunicação do centro de dados até aos locais de acesso remoto.

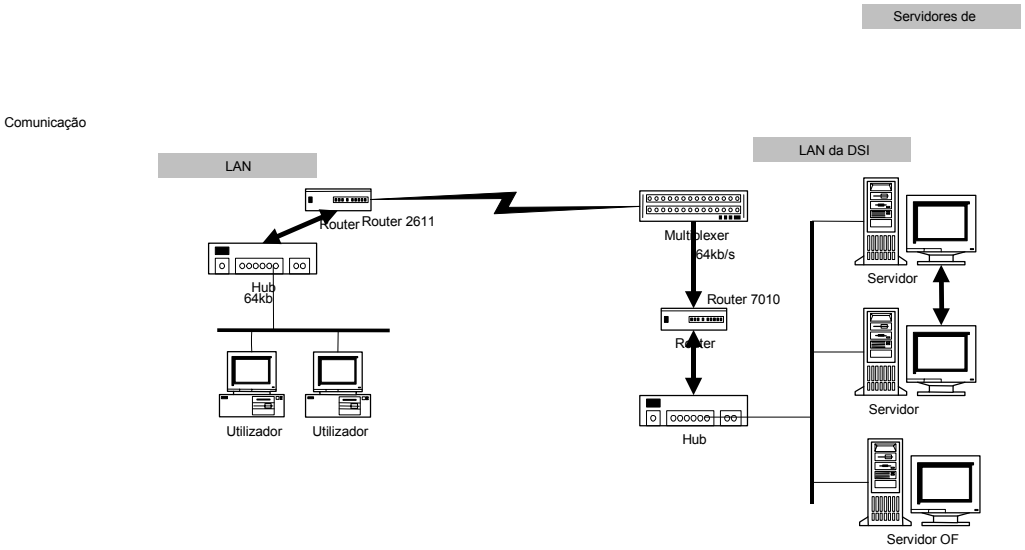


Figura 5.4. Comunicação entre o centro de dados e um local remoto

Os servidores de base de dados e aplicação estão apresentados na tabela em baixo, incluindo as suas características.

Servidor	Modelo	Base de dados/Aplicação	Process.	Memória
mpecu000	hp9000/899/k570	Base de dados/Aplicação	4	3.75 GB
mpecu001	hp9000/898/k370	*Base de Dados Standby*	2	1.5 GB
mpecu004	hp9000/898/k370	Aplicação Girafe	4	1.0 GB
mpecu005	hp9000/800/N4000-55	Base de dados	4	6.0 GB

Tabela 5.2. Características dos servidores

O *switch* central é de 96 portas com uma taxa de transferência de 100MBits/s.

Os canais de comunicação de dados são baseados na comunicação via satélite, comunicação via *multibit*, e a comunicação via fibra óptica. A largura de banda varia entre 64 K a 128 K como mostra a tabela 5.3.

Local	Largura de banda	Taxa de transferência
Maputo (IFT, Oficinas, Polana), Xai-xai, Beira, Pemba, Nampula, Inhambane	128K	Entre 12 a 25K/s (download da internet)
Chimoio, Tete, Nampula, Nacala, Quelimane,	64K	Sem informação
Lichinga	48.0 K	Sem informação

Tabela 5.3. Largura de banda dos locais de acesso remoto

O interface do cliente (utilizador) é um computador pessoal (PC) com o sistema operativo *windows*.

Na aplicação Girafe, é configurado no interface do cliente o protocolo *telnet* que corre no sistema operativo dos servidores de base de dados e aplicação. Na aplicação OF o acesso à aplicação é baseado na tecnologia *web*. Em ambos os casos, a comunicação é de um para um e assenta no protocolo TCP/IP.

A dificuldade da rede da TDM verifica-se nos pontos de acesso remoto onde mostra uma latência relativamente elevada e menor capacidade na detecção de falhas no hardware da rede o que muitas vezes é reportado via telefone depois da falta do serviço.

As implicações da ausência de mecanismos de detecção de falhas de hardware são a demora no diagnóstico e recuperação da falha que influencia a disponibilidade e o desempenho das aplicações. Este problema afecta as relações de dependência entre a TDM e os clientes externos, principalmente.

Capítulo VI

6. Análise e discussão

Neste capítulo discute-se o estado actual da empresa relativamente à tolerância a faltas dos seus sistemas de informação e, com base na análise dos valores apresentados, são feitas propostas de alteração que se julgam apropriadas para corrigir os problemas detectados. Na prática, verifica-se que as falhas de hardware e de *software* ocorrem com alguma regularidade, o que se traduz em frequentes faltas dos serviços em locais isolados a prestar por esta empresa. Nota-se no entanto que as falhas dos servidores são raras, sendo por vezes causadas pela falha de energia ou de *hardware* das comunicações.

Chama-se a atenção ao leitor para o facto do cálculo da disponibilidade e fiabilidade ter sido efectuado com algumas limitações, pois nem todas as falhas são registadas no *help desk*, facto agravado pela ausência de acesso aos registos dos *routers* (*logfiles*) ou dos locais de acesso remoto. Por essa razão, os valores para acesso por Internet e alguns locais de acesso remoto apresentam disponibilidade e fiabilidade melhores do que se verifica na realidade. Em relação ao desempenho não foi possível a sua quantificação por falta de mecanismos precisos de avaliação de *throughput*, que inclui a taxa de transferência de dados na rede e o comprimento dos pacotes (latência). Devido à grande dispersão geográfica, também não foi possível efectuar deslocações aos locais remotos, tendo sido apenas medidos tempos de "ida e volta" de pacotes (*ping*).

6.1. Probabilidade da falha

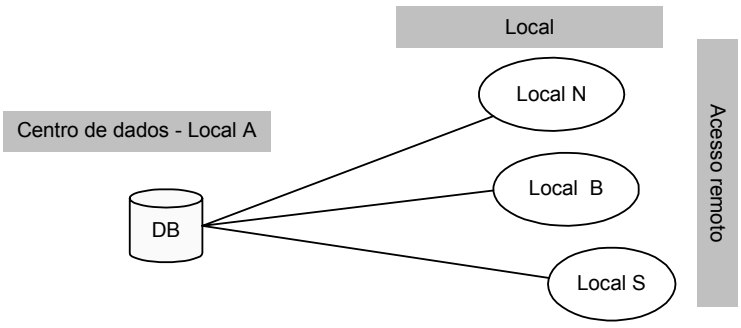


Figura 6.1. Centro de dados e os locais de acesso remoto

Seja o ponto A o centro de dados, Maputo, onde residem os servidores. N, B e S três áreas arbitrárias de acesso remoto.

As probabilidades de falha no Local A das aplicações OF, Girafe e a Internet são dadas pelo gráfico da Figura 6.2.

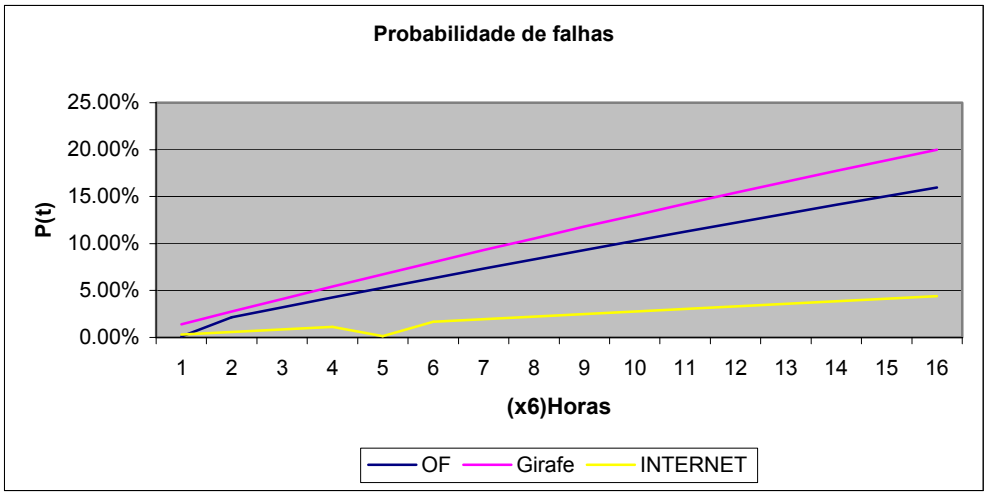


Figura 6.2. Gráfico da probabilidade de falha no centro de dados

A probabilidade de falha em todas as aplicações aumenta com o tempo. Em termos comparativos a probabilidade de falha é elevada e atinge valores superiores a 5% [6,9].

A falha em A afecta todos os outros locais. A probabilidade de falha em N, B e S é ainda mais elevada devido a falha da rede de comunicação nesses pontos. A figura 6.3 mostra o comportamento da probabilidade de falha em função do tempo da aplicação Girafe. Foram considerados quatro locais de amostra, nomeadamente, Lichinga, Nacala, Beira e Inhambane.

A figura 6.3. mostra a probabilidade de falha nos pontos de acesso remoto (N,B e S).

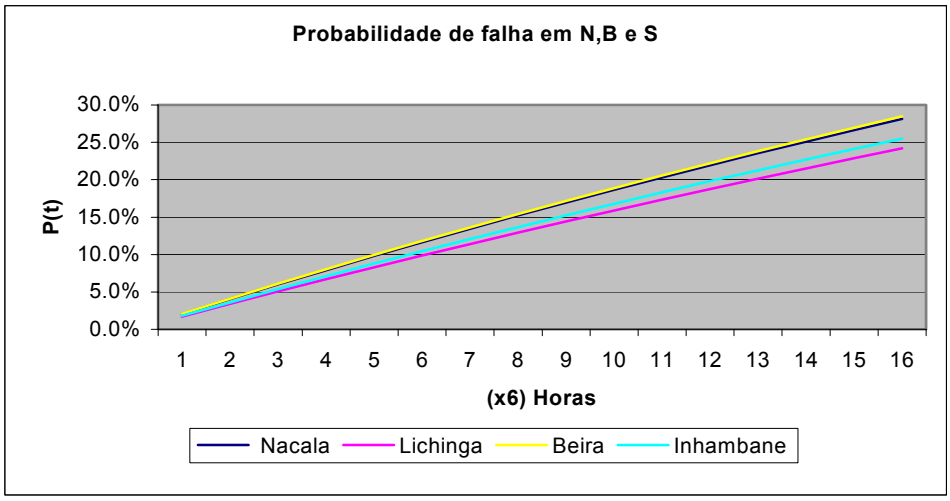


Figura 6.3. Probabilidade de falhas em N, B e S

As falhas que ocorrem quase em todos os locais de acesso remoto são do tipo *fail-stop* e são identificadas pela ausência de serviços. Quanto à natureza das falhas pode se identificar dois grupos de falhas. O primeiro grupo corresponde às falhas possíveis de recuperar. É este grupo de falhas que ocorrem com regularidade na empresa, mas que resultam num menor impacto nos processos da organização. Porém este grupo de falhas origina maiores dificuldades na disponibilidade de serviços devido a falta de mecanismos de detecção. Frequentemente a falha é identificada pela falta de serviços ou depois do utilizador telefonar. A detecção de falhas utilizando o telefone aumenta custos à empresa para além de piorar a disponibilidade.

O segundo grupo de falhas corresponde àquelas que é impossível de recuperar (fogo, sabotagem e cheias). A nível da empresa não existe medidas de protecção contra este grupo de falhas e nem mecanismos de recuperação. São sinais da falta de protecção contra estas falhas, a centralização dos servidores num mesmo local e as salvaguardas nas bandas magnéticas armazenados no mesmo edifício. Até ao momento, a empresa nunca sofreu este tipo de falhas, mas os acontecimentos externos e as condições actuais de armazenamento de dados exigem outras medidas de prevenção.

Um método básico de tolerância a falhas num sistema informático é a redundância dos seus componentes. Assim, para tolerar k falhas é necessário dispor de k+1 componentes com falhas independentes [6], o que não se implementa na TDM. Por exemplo, a probabilidade de

falha na aplicação OF durante 72 horas no centro de dados (A) é de cerca de 0.122. Se a TDM utilizasse uma redundância de dois servidores com a mesma probabilidade de falha, iria diminuir a probabilidade para $P(t)=(0.122)^2 =0.0148$. Quer dizer, a probabilidade de pelo menos um servidor estar a funcionar seria igual a 0.9852 em 72 horas. Este valor supera 0.878 da fiabilidade de um único servidor.

6.2. Disponibilidade

Nas figuras seguintes mostram-se os valores para a disponibilidade e fiabilidade calculados a partir dos dados recolhidos no *help desk*, *logfiles* e observações feitas durante os meses de Maio e Junho de 2002. A amostra é do primeiro semestre de 2002 para a aplicação Girafe e até Maio para a aplicação OF. Conforme se vê na figura 6.4, a disponibilidade da aplicação Girafe no entro de dados, é de cerca de 97.90%. Significa que 2.1% da disponibilidade na aplicação Girafe perde-se pelas falhas de energia, falhas da base de dados, salvaguardas mensais, manutenção e processamento de facturação mensal.

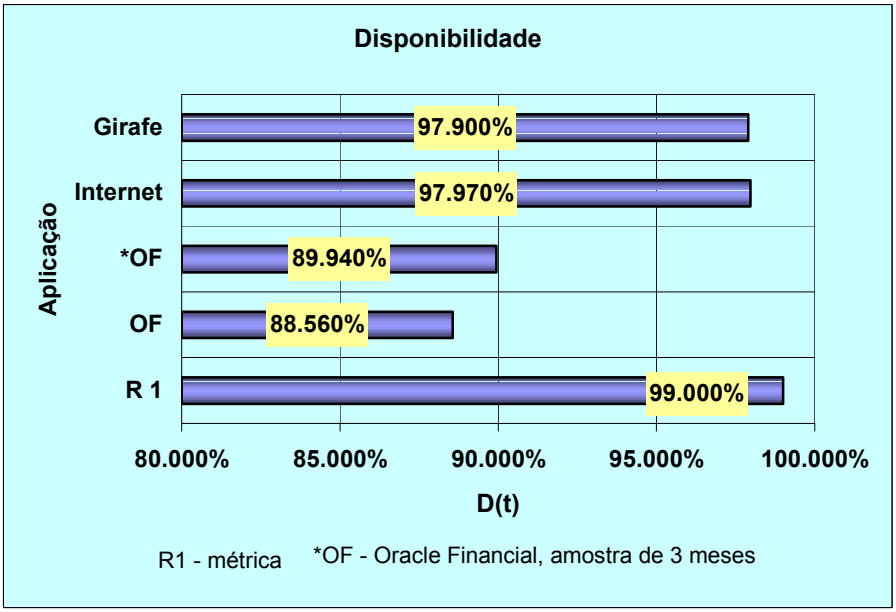


Figura 6.4. Gráfico da disponibilidade

Disponibilidade e Fiabilidade

	Probabilidade			Disponibilidade			Fiabilidade		
LOCAL	OF	Girafe	Internet	OF	Girafe	Internet	OF	Girafe	Internet
Centro de dados	12.2%	15.4%	3.3%	88.6%	97.9%	97.9%	87.8%	84.6%	96.6%
Xai-xai	14.1%	18.9%	6.6%	88.1%	96.9%	97.0%	85.9%	81.8%	93.4%
Inhambane	16.0%	19.7%	8.2%	88.1%	96.8%	96.7%	84.0%	80.3%	91.8%
Beira	21.3%	22.2%	9.7%	88.5%	97.8%	97.8%	78.7%	77.8%	90.3%
Chimoio	16.1%	18.3%	6.6%	88.2%	97.6%	97.7%	83.9%	81.7%	93.4%
Nacala	19.0%	21.9%	8.6%	81.7%	90.0%	94.5%	81.0%	78.1%	91.4%
Pemba	15.8%	21.6%	8.3%	80.3%	91.3%	94.8%	84.2%	78.4%	91.7%
Lichinga	16.6%	18.8%	8.3%	77.4%	92.7%	95.3%	83.4%	81.2%	91.7%

Tabela 6.1. Disponibilidade e fiabilidade

Durante o mesmo intervalo de tempo, a disponibilidade na aplicação OF atinge em média 88.56%, perdendo também cerca de 11.4% da disponibilidade nas falhas de energia, falhas da base de dados, salvaguardas mensais, testes de novos produtos e *upgrade* de *hardware* ou *software*. A disponibilidade da Internet no centro de dados é de cerca de 97.97%. Também perde 2.03% da disponibilidade por causa de falha de *hardware*, manutenção e falta de serviços (energia e telecomunicações).

Em relação aos locais de acesso remoto, a disponibilidade é ainda mais baixa quando comparada com o centro de dados, devido à falhas de *hardware* das comunicações e de energia. A disponibilidade nos locais de acesso remoto atinge valores relativamente baixos (Nacala, Pemba e Lichinga) em todas as aplicações devido a falhas da rede, falta de energia local e o elevado tempo que separa a participação da avaria e a respectiva reparação. Por exemplo, a avaria e recuperação de um *router* nos pontos de acesso remoto depende do deslocamento de um técnico de Maputo até ao local.

Um sistema de alta disponibilidade deve ter valores próximos de 100% (99.999%) [6,9,34,35] e com um bom tempo de resposta. Na TDM, a disponibilidade em todos os sistemas é relativamente baixa e o tempo de resposta é elevado no Centro e no Norte de Moçambique.

6.3. Fiabilidade

Os mecanismos de tolerância a falhas que a empresa implementa são a nível de energia, dos servidores e a nível do armazenamento de dados (bandas magnéticas). A fiabilidade é influenciada por falhas tanto dos sistemas como dos serviços de energia e telecomunicações. O método de detecção e comunicação das falhas na empresa é um processo demorado feito via telefone prejudicando a disponibilidade. A fiabilidade foi calculada para os sistemas funcionarem durante 72 horas sem parar. A escolha deste tempo foi baseada no comportamento das funções de distribuição das falhas apresentado no anexo A. A fiabilidade para a aplicação

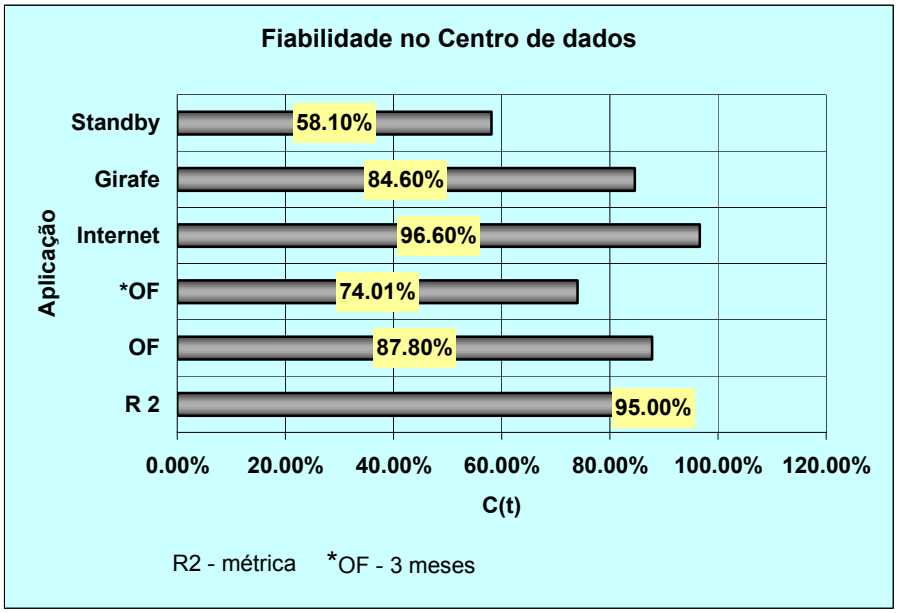


Figura 6.5. Gráfico da fiabilidade

Girafe no centro de dados é igual a 84.6%, valor que é determinado tomando em consideração as falhas de energia e da base de dados. A fiabilidade é relativamente baixa nos pontos de acesso remoto devido a falhas da rede de comunicação e da energia local.

A tabela 6.1 e a figura 6.5 apresentam o quadro geral da fiabilidade e a probabilidade de falha dos sistemas.

Em relação ao servidor de espera da base de dados de produção Girafe, a fiabilidade é de cerca de 58.1%, valor considerado baixo se se considerar que o sistema deve funcionar 24 horas,

7 dias por semana sem parar. As causas de baixa fiabilidade do servidor *standby database*, pode estar associado ao mecanismo de transferência dos *datafiles* entre o servidor da base de dados de produção para o de espera. Esta transferência é efectuada por um programa, elaborado em linguagem "*shell"* do Unix. O sistema não é fiável, falhou 20 vezes em seis meses, com uma duração média de 84.25 horas, valor considerado elevado para segurança de dados. Um método de transferência de dados baseado na replicação configurada com base nos produtos da *Oracle* poderá certamente reduzir este tempo de falhas.

Em relação a OF a confiabilidade no centro de dados é de cerca de 87.8%. Quer dizer, a probabilidade do sistema funcionar durante 72 horas sem falhar é igual a 0.878. Este valor foi determinado tomando em consideração as falhas de energia e da base de dados. A confiabilidade é comparativamente baixa nos pontos de acesso remoto devido à falha de *hardware* das comunicações e energia. A Internet tem uma confiabilidade relativamente elevada em comparação com as duas aplicações.

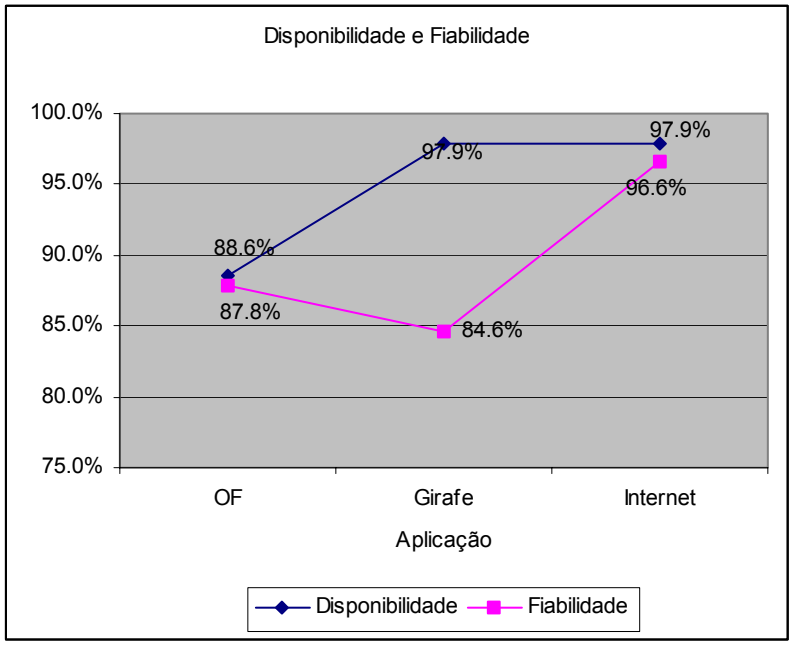


Figura 6.6. Gráfico da disponibilidade e confiabilidade no centro de dados

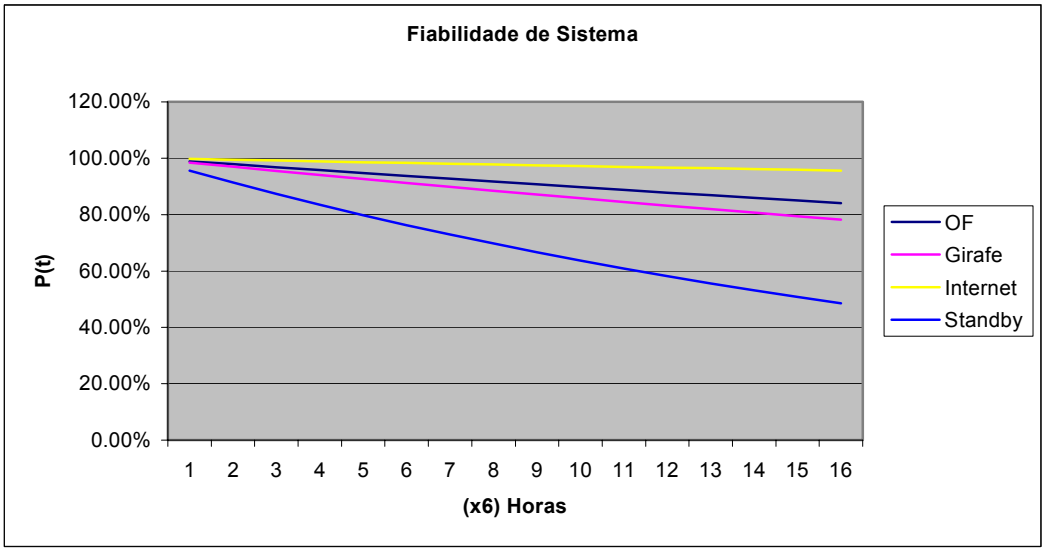


Figura 6.7. Gráfico de confiabilidade de cada sistema

A figura 6.7 mostra que a fiabilidade diminui com o tempo atingindo valores mais baixos nas aplicações OF, Girafe e servidor da base de dados de espera ou *standby Database*. A Internet tem uma melhor fiabilidade comparada com as outras aplicações, valor superior a 95% em 72 horas. O valor da Internet tem fiabilidade relativo superior quando comparada com as outras aplicações porque o registo das falhas não foi consistente comparada com as bases de dados, tendo se baseado em fontes de gestão. Enquanto que nas bases de dados para além do registo do *help desk* foi possível analisar os *logs* das bases de dados. As aplicações OF e Girafe por sua vez tem valores que se aproximam devido a partilha da falta de energia e de rede de comunicações.

6.4. Desempenho

O desempenho dos SI na TDM depende da disponibilidade dos recursos do servidor e da rede. Como o sistema é centralizado, o desempenho poderá ser afectado quer por uma eventual sobrecarga ou saturação do servidor, que se traduz por tempo de espera até à transacção ser escalonada para execução, quer por atrasos causados por linhas de comunicação lenta. A figura 6.8 mostra o comportamento do tempo de resposta por local de acesso.

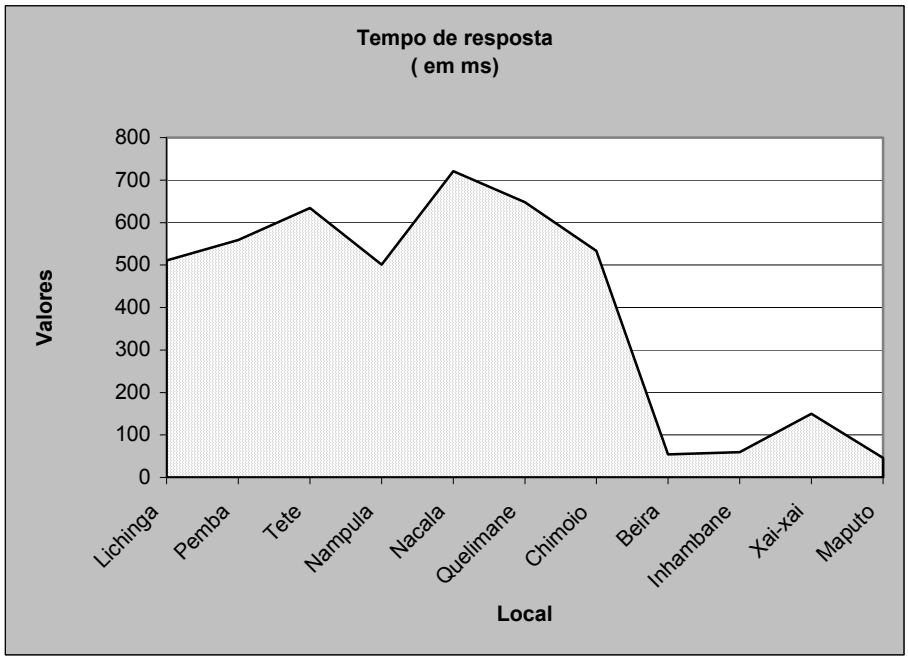


Figura 6.8. Tempo em milisegundos de resposta nos pontos de acesso remoto

Verifica-se que nos pontos de acesso remoto o tempo de resposta cresce gradualmente tendo valores elevados no Centro e Norte de Moçambique (Chimoio, Nacala, Nampula, Tete, Lichinga e Pemba). Estes testes medem a latência desde Maputo até ao interface do *router* na zona Centro e Norte. Note-se que os verdadeiros valores do tempo de resposta são bastante superior aos apresentados no gráfico pois incluem factores tais como a configuração da rede local que inclui a largura da banda, *hardware* envolvido (*hubs*) e os protocolos de comunicação.

O elevado tempo de resposta é visível nas próprias aplicações pois estas apresentam uma latência superior a um segundo entre a fonte primária em Maputo e o utilizador local. As causas

possíveis de elevada latência nos pontos de acesso remoto são a centralização de dados, a comunicação em grupo baseado no *broadcast* que origina maior tráfego na rede incluindo a utilização de *hubs* nos locais de acesso remoto. Para além dos factores já mencionados pode-se identificar na empresa dois métodos de comunicação, o primeiro baseado na comunicação via satélite e o segundo na fibra óptica. Na prática os dois meios de comunicação produzem tempos de respostas diferentes sendo de bom desempenho a fibra óptica, por exemplo Beira. Outros factores que podem contribuir para a lentidão das aplicações nos locais de acesso remoto relacionam-se com a maneira como são acedidos os dados. A aplicação Girafe (BD dos clientes) utiliza o protocolo *telnet* e FTP com a desvantagem adicional de baixa segurança de dados na rede. A aplicação OF utiliza a WEB. Estes dois métodos de acesso a dados também condicionam tempos diferentes de resposta.

A figura 6.9. mostra graficamente os protocolos de comunicação de uma LAN.

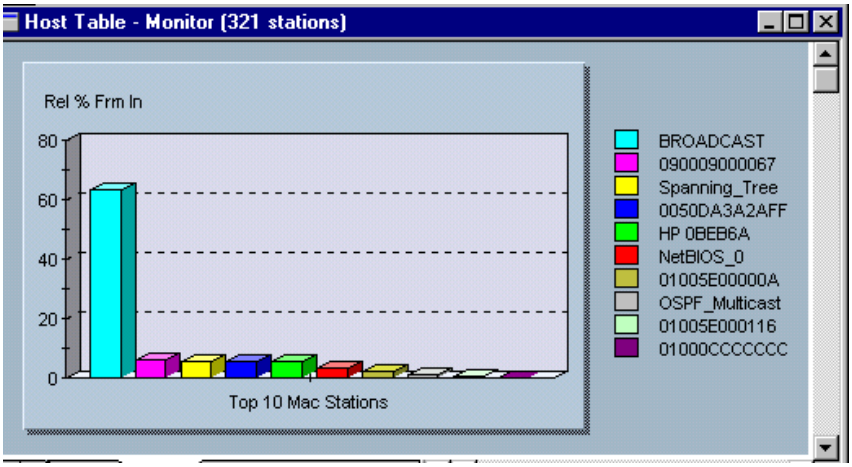


Figura 6.9. Protocolos de comunicação de uma LAN

6.5. Alternativas de distribuição de dados replicados

Os Sistemas de Informação da empresa tratam os processos do negócio em tempo real e os dados constituem o centro de todas as operações. A sua falta torna impossível a prestação de serviços. Esta dependência dos clientes em relação a SI (BDs) exigem da empresa a adopção de medidas que garantem a disponibilidade e minimizam as falhas. Nos pontos de acesso remoto, uma réplica local pode aumentar a disponibilidade e o desempenho.

Quando o objectivo principal da replicação é aumentar a tolerância a falhas e facilitar a recuperação de desastres, é aconselhável replicação de servidores em locais geograficamente distintos. Uma replicação passiva composta por um servidor primário e um servidor secundário é uma técnica viável, não obstante os custos de *hardware* e recursos humanos para a administração e manutenção. Os outros custos são indirectos e causados pela complexidade de actualização do estado do secundário.

Se se considerar o objectivo final de aumentar a disponibilidade e também o desempenho, então deverá implementar-se a replicação com separação de dados em todos os locais onde são necessários, isto é, a distribuição das réplicas por todos locais. Este método de replicação é largamente utilizado nos sistemas bancários e de bordo devido à sua natureza crítica.

Para se obter bom desempenho e alta disponibilidade, os sistemas replicados são normalmente baseados em dois ou mais servidores virtuais chamados *clusters* com falhas independentes. Os *clusters* também podem ser utilizados para a tolerância a falhas, mas não são eficazes para as falhas catastróficas. Os *clusters* oferecem enormes facilidades de uma base de dados incluindo as trocas virtuais do primário para o secundário sem interromper a exploração, nomeadamente, *backups* e manutenção em *online*. As estratégias de replicação constituem o centro de qualquer plano de recuperação da falha. Uma boa replicação deve incluir alta disponibilidade de dados, distribuição consistente da informação, controlo administrativo centralizado fácil, autonomia local (independência) e acesso a dados em fontes heterogéneas.

6.6. Análise das soluções

Nas secções seguintes são apresentadas e estudadas em detalhe três soluções alternativas para a replicação de dados e serviços da empresa TDM:

- Replicação síncrona Maputo-Matola
- Replicação activa Maputo-Beira
- Replicação com servidor virtual

6.6.1. Replicação síncrona passiva envolvendo dois pontos da zona sul (Maputo-Matola)

Uma replicação passiva de dados envolvendo dois locais da zona sul (Maputo e Matola) afigura-se como uma solução económica para a empresa principalmente em relação a custos de manutenção e administração. É uma solução que pode ser implementada mesmo recorrendo ao *mirror* de discos em locais separados sem necessidade de utilização dos servidores, bastando para isso construir uma rede de comunicação fiável, por exemplo a fibra óptica. Esta solução é tecnicamente viável somente para falhas de local (exemplo fogo), mas não resolve problemas do desempenho e alta disponibilidade. Uma replicação baseada apenas em duplicação de discos pode ainda ser prejudicial para a empresa porque os discos sem servidor não permitem a venda de serviços.

Uma solução de tolerância a falha utilizando servidores replicados tem como grande vantagem a capacidade de se poder configurar o servidor secundário para atender os clientes em caso de falha do primário, ganhando assim uma melhor disponibilidade. Nas organizações com a replicação baseada em esquemas primário/secundário, nos tempos de bom funcionamento do servidor primário o secundário poderá ainda ser usado para a realização de outras tarefas.

Embora a replicação baseada em *mirroring* seja mais económica quando comparada com a replicação do tipo primário/secundário, para o caso em estudo esta última seria uma solução mais apropriada.

A solução de replicar os dados entre Maputo e Matola (*mirror* de dados) é fundamental para a segurança de dados da empresa que até agora está garantida pelo mecanismo dos *backups* nas bandas magnéticas e localizados no mesmo local. A sincronização e actualização do secundário na Matola recomenda-se que seja passiva e executado num intervalo de tempo não superior a 15 minutos. A replicação passiva vai permitir melhor tempo de resposta para os clientes uma vez que o servidor primeiro vai dar resposta ao cliente e só mais tarde vai actualizar o estado do secundário. O intervalo de tempo considerado tem por sua vez a vantagem de minimizar a quantidade das transacções perdidas nos eventos de falha do primário. A implementação da replicação deve ser feita utilizando os produtos da *oracle* assumindo que o primário e o secundário tenham as mesmas características e configurações. Os clientes devem estar configurados a nível aplicacional com dois endereços, um primário e um secundário. Este

tipo de redundância de acesso para os clientes também pode ser feita utilizando *hardware* próprio que inclui uma máquina de estado e portas de *routers* a apontarem para dois endereços diferentes um primário e outro secundário. A outra solução mais simples consiste na mudança do *IP* do secundário e atribuir a do primário. Estes procedimentos terão menor demora nos clientes comparada com o estágio actual.

Padece no entanto de algumas limitações, nomeadamente:

- persistência de elevado tempo de resposta nas zonas Centro e Norte de Moçambique.
- dependência das redes de comunicação de dados (elevados custos da rede) e fraco desempenho.
- centralização de dados no mesmo local (primário ou secundário) pode originar o congestionamento das redes de comunicação.
- As comunicações para o Centro e Norte do País a partir da Matola são limitadas porque dependem do centro de transito em Maputo.

6.6.2. Replicação activa envolvendo a zona sul e centro (Maputo – Beira)

Para proteger todas as falhas e elevar a disponibilidade de dados aconselha-se uma replicação envolvendo dois centros de dados, Maputo e Beira conforme ilustrado na figura seguinte.

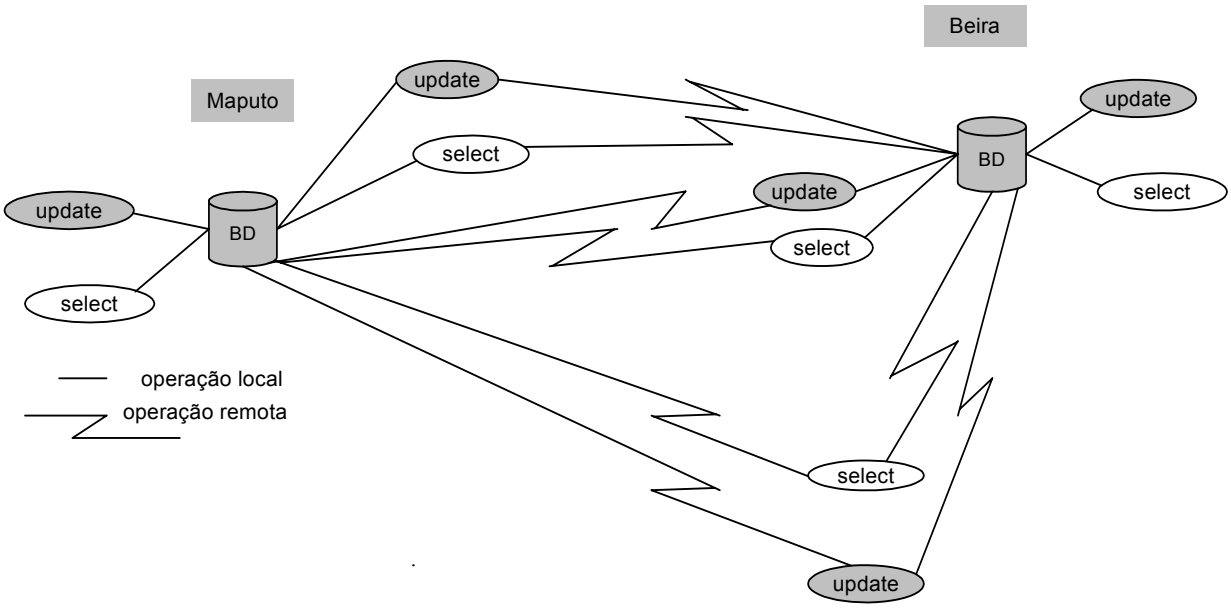


Figura 6.10. Replicação activa entre Maputo e Beira

Uma réplica na Beira pode ainda ser aproveitada para a harmonização de processos organizacionais, equilíbrio da carga computacional e melhoramento da qualidade de serviços. Esta estratégia pode diminuir os custos da rede de comunicação de dados e uma recuperação de dados mais eficiente se os dados forem particionados por localização física. Cada local, Maputo ou Beira, pode ser tomado como grupo de recuperação e os *backups* podem ser planificados como parte de cada local no fim de cada processamento.

O tipo de replicação a implementar depende dos recursos económicos necessários e dos objectivos preconizados. Uma replicação activa para a tolerância a falhas permite que os dados sejam sempre consistentes (consistência rígida), mas devido a separação dos dois centros de dados o único mecanismo de manutenção de consistência total terá de se basear nos protocolos ROWA. Note-se que a replicação activa requer o uso de protocolos de *commit* global para coordenar as unidades distribuídas de trabalho, e que o desempenho das aplicações se pode degradar devido ao uso dos *locks* nos gestores de recursos associado aos canais de comunicação.

A replicação activa é dispendiosa para a TDM sobretudo nas operações de actualização da base de dados e na ocorrência de falha porque pode afectar o desempenho e tornar o sistema

lento. Enquanto que a replicação passiva manterá uma latência constante entre o primário e o secundário. A metodologia de replicação passiva tem ainda as vantagens de transparência para os utilizadores porque não requer os *locks* nos gestores de recursos, mas a consistência é relativamente fraca. A escolha da replicação passiva para a TDM tomou em consideração as falhas, tempo de resposta e consistência. O candidato considera que o sistema raras vezes vai falhar, e a empresa vai implementar a redundância de canais de comunicação tal como é apresentado neste trabalho. O intervalo de actualização do estado entre os dois servidores deve ser menor possível, ordem dos minutos. Esta medida vai reduzir o impacto das transacções perdidas em caso de falhas de um dos gestores e a necessidade de utilização do secundário.

A arquitectura apresentada consiste em dois gestores de recursos, um servidor em Maputo e outro na Beira. O servidor da Beira funcionará como o primário para os clientes locais e visto como secundário para os clientes de Maputo. O servidor de Maputo funcionará como primário para os clientes locais e visto como secundário pelos clientes da Beira. A nível da empresa com excepção da aplicação Girafe todas as outras são de acesso via *browser*. Esta facilidade permite uma configuração redundante dos interfaces de acesso a dados dos clientes, obedecendo prioridades de cada local. No estado normal de funcionamento os clientes, estarão conectados ao único servidor primário local. Quando acontece uma falha que obriga a utilização do secundário o cliente vai indicar no *browser* o novo primário (secundário).

Em relação a aplicação Girafe uma replicação completa (eficiente) da base de dados requer a migração da versão de *oracle* 7.3.4.5.0 para uma versão superior a 8. Pois a versão actual de *oracle* permite apenas a replicação simples limitando os objectos a replicar na base de dados.

As vantagens desta metodologia de replicar os dados são: transparência para os utilizadores, melhora a disponibilidade e tolerância a falhas. Os processos de *backups* e recuperação serão mais rápidos uma vez que serão feitos em *online* ou seja, cada local será tomado como *backuup* de outro permitindo uma melhor planificação. A solução descrita é adequada para a TDM uma vez que permite outras facilidades que incluem a expansão do negócio, escalabilidade, uniformização de processos organizacional, equilíbrio da carga computacional incluindo o desempenho.

6.6.3. Replicação para a disponibilidade – servidor virtual (*cluster*)

Um servidor virtual é composto por uma máquina de estado ou dispositivo de armazenamento redundante do estado, um primário e um secundário. A máquina de estado guarda os estados do primário, quando o primário falha transfere o estado para o secundário e passa a responder os pedidos dos clientes. O estado pode ser centralizado ou distribuído.

Um *cluster* traz vantagens para a alta disponibilidade e reduz o tempo de recuperação depois da falha com baixos custos de administração e manutenção. Devido a configuração dos *clusters*, próximo um do outro estes sofrem de limitações de distância o que permite serem vulneráveis a falhas catastróficas. A outra limitação dos *clusters* na TDM é a contínua fraca distribuição da carga de dados para a zona Centro e Norte. A solução dos *clusters* coloca a empresa requisitos complementares de maior largura de banda e redundância de canais de comunicação para elevar a disponibilidade. Esta solução é necessária sobretudo na implementação de serviços baseadas no comercio electrónico uma vez que os *clusters* são capazes de garantir a disponibilidade correspondente à exigência dos clientes. Por exemplo, a ausência deste tipo de serviços obriga os clientes da TDM a utilizar telefone ou deslocar-se à TDM para obter informação ou realizar novas transacções.

Esta solução deve ser vista como forma de resolver o problema de alta disponibilidade na TDM e não é abrangente por isso que continua ser aconselhado a solução anterior por ser completa e abrangente para os problemas da empresa.

6.7. Rede de comunicação de dados e o desempenho

A rede de comunicação de dados da TDM apresenta uma elevada latência e probabilidade de falha no Centro e no Norte de Moçambique. Como resultado, as aplicações são lentas. As causas de elevada latência na rede podem ser:

- Consequência directa da centralização de dados que origina um tráfego elevado na rede. Os canais de comunicação têm uma capacidade limitada e quando o tamanho de dados ou pacotes se aproxima da sua capacidade total dá origem a uma fila de espera no *host* emissor e nos locais intermediários. A transmissão de pacotes pode experimentar dificuldades (ou então ser bloqueada pelo tráfego). Se a transmissão continua no mesmo sentido a fila de

espera cresce e atinge o limite do espaço disponível no *buffer*. Como resultado, o local receptor (visitado) deixa de receber outros pacotes enviados pelo *host* emissor. Este problema pode dar origem a perdas na rede que é compensado pelos mecanismos de verificação e retransmissão (*checksum*, *bit* de paridade). Mas quando a taxa de perdas de pacotes atinge um nível substancial, o seu efeito no *throughput* da rede é destrutivo. Em regra um carregamento de um *link* da rede que atinge 80% da sua capacidade, o *throughput* total tende a perder-se como resultado da perda de pacotes, a não ser que a utilização de carregamento de *links* pesados seja controlado [9].

- Na comunicação via satélite os tempos de ida e volta são da ordem dos 600 ms em particular para as zonas de acesso remoto. O elevado tempo de resposta pode ser resultado do enfraquecimento do sinal nos pontos intermediários. A distorção do sinal eléctrico acontece na comunicação *multibit,* tecnologia em uso na TDM*,* onde o sinal atravessa vários condutores. As distorções e ruído do sinal eléctrico podem provocar perdas de pacotes na rede. Uma solução alternativa é a introdução de comunicação via fibra óptica em todos os locais, o que não é económico para a empresa. A segunda solução alternativa é criar centros intermédios (fila de espera) no Centro. No estágio actual a comunicação Maputo - Beira tem um bom desempenho. Uma réplica na Beira pode diminuir a quantidade de dados a serem deslocados entre os dois centros.

- Para evitar o único ponto de falha, uma redundância do canal de comunicação pode criar condições de uma disponibilidade assinalável. Habitualmente, as empresas com dois centros de dados implementam a configuração dinâmica dos seus clientes ou adquirem dispositivos de automatização das falhas para alternar o sentido da replicação na eventualidade de uma falha resultando numa elevada disponibilidade. Apesar de serem viáveis, estas técnicas tem como desvantagem o custo de aquisição e manutenção.

6.7.1 Proposta de arquitectura da rede de dados da TDM

A elaboração do plano de recuperação de falhas incluindo a redundância de canais deve ser encarada pela gestão da empresa como uma oportunidade para eliminar ou minimizar erros humanos e decisões improvisadas tomadas pelos técnicos inexperientes ou executada em situação de *stress*. A elaboração de plano de recuperação da TDM deve ser baseado em vários cenários para ser abrangente e flexível na recuperação de falhas.

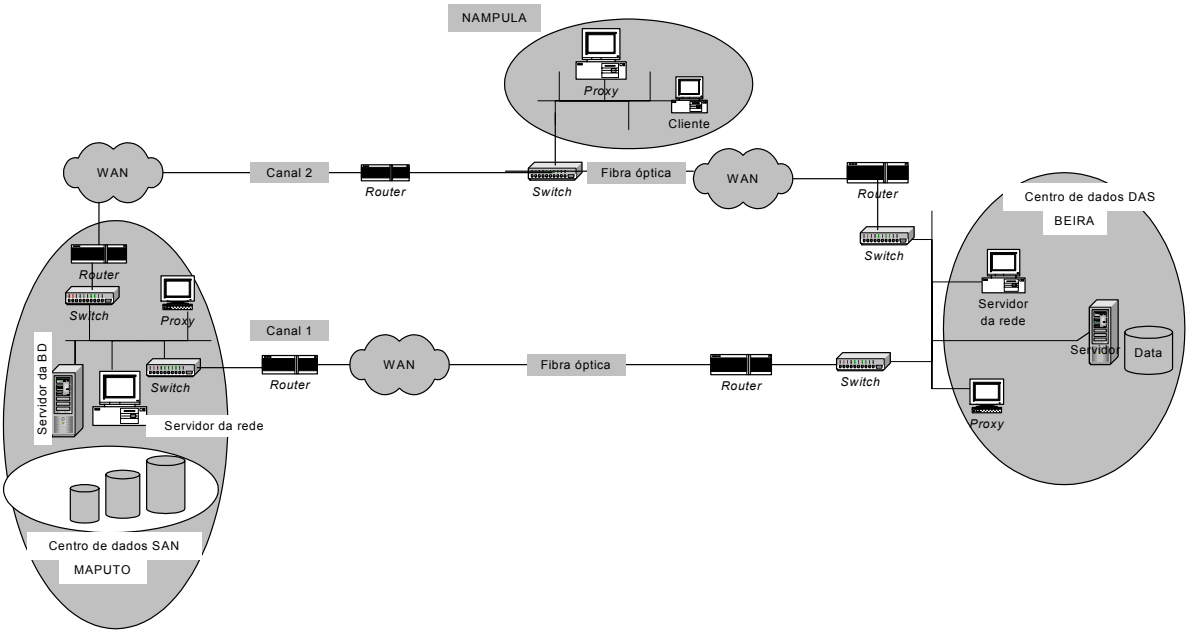


Figura 6.11. Arquitectura da rede de dados da TDM

O método que a seguir se descreve para implementar a recuperação das falhas da rede, e sobretudo aumentar a tolerância a falhas e o desempenho, consiste em canais de comunicação redundante e cache distribuído. A redundância dos canais de comunicação, que se designa por Circuitos Permanentes Virtuais duais (PVC's) dever-se-ia implementar numa primeira fase em locais críticos como Beira devido à proposta de inclusão neste local de um centro de dados e Nampula para efeitos do equilíbrio da carga computacional na zona Norte do País. Uma *cache* distribuída ou replicação via *cache* nos locais críticos indicados para melhor partilhar os dados com melhores tempos de resposta.

Recomenda-se a utilização do método baseada na *web* (*browser*) para aceder todas as aplicações da empresa. Nesse caso os canais de comunicações devem ter uma largura de banda necessária para diminuir a latência.

Uma arquitectura possível para a rede da TDM, ilustrado na figura 6.11 (e que se apresenta com mais detalhe no Anexo B), consiste na existência de três locais estratégicos de redundância de canais, Maputo, Beira e Nampula, respectivamente. Esta arquitectura permite estabelecer duas rotas de comunicação para o mesmo local. Por exemplo, a comunicação Beira - Maputo pode ser feita directamente utilizando o canal 1 ou recorrendo ao canal 2 Maputo – Nampula – Beira, implementando endereços diferentes. Para além da tolerância a falhas, a existência de rotas múltiplas passaria a ser um factor regulador de trafego. A implementação desta proposta de solução para a TDM requer a inclusão de um servidor da rede com *software* específico de gestão e monitorização dos canais para detectar falhas.

Os PVC podem ser implementados com uma configuração de ambos os canais permanentemente activos ou um canal activo e outro em espera. No primeiro caso uma redundância de canais na TDM tem vantagens porque vai evitar sobrecarga da rede e aumentar a disponibilidade e desempenho implementando canais de acesso múltiplos. O segundo caso é económico uma vez que o canal de redundância pode ser de menor largura da banda e pode ser activado somente quando ocorrer falha e manualmente. Este método de redundância utilizando o canal de salvaguarda inactivo tem ainda a vantagem quando a empresa não tem capacidade técnica para configurar uma redundância activa podendo ser activado manualmente. Mas, como a empresa está numa fase avançada da implementação da fibra óptica a Direcção dos Sistemas de Informação da empresa pode implementar PVC dual activos com configurações permanentes nos locais acima referidos.

A solução apresentada implica custos na aquisição de *hardware* de comunicações e formação de técnicos a nível local para a manutenção. Mas comparada com o estágio actual, é uma proposta viável porquanto a empresa já tem circuitos permanentes que ligam Maputo - Beira e Maputo - Nampula assim como muitos outros locais.

Os dados recolhidos mostram que o custo de manutenção do *harware* da rede no Centro e Norte do país varia entre os 3 mil a 5 mil dólares/ano e o custo de um *router* e *switch* tem o custo estimado entre 3 mil a 4 mil dólares (informação de 2002). A avaria do *hardware* das

comunicações na empresa tem outros custos indirectos associados como por exemplo manter parte dos utilizadores sem produzir, paralização da venda de serviços da empresa e pagamento dos fornecedores. Estes factores todos aumentam o custo e justifica-se a implementação desta proposta.

6.8. Distribuição e partição de dados na replicação

A replicação de dados requer uma identificação selectiva de dados a replicar e a maneira como distribui-los. No caso da TDM, pode se considerar duas estratégias de replicação, a replicação total e a replicação parcial. Uma replicação total da base de dados tem vantagens na disponibilidade e desempenho, em particular na execução de interrogatórios à base de dados, porque o resultado pode ser obtido localmente. Mas uma replicação total da base de dados pode tornar os mecanismos de controlo de concorrência e recuperação mais dispendiosos para além de as operações de actualização poderem ser lentas.

Uma segunda alternativa de distribuição de dados designada por replicação parcial consiste na colocação de alguns fragmentos de dados nos locais onde são necessários. A selecção de dados a replicar depende do desempenho, disponibilidade do sistema e do tipo das transacções submetidos em cada local. Na TDM a disponibilidade e o desempenho constituem um requisito indispensável e as transacções são frequentemente submetidas em todos os locais e acedem dados particulares (fragmentação horizontal de dados), então o conjunto desses dados na base de dados podem ser colocados somente nesse ponto.

A partição da base de dados pode ser feita ou na fonte primária ou na secundária. Os dados particionados na fonte primária têm vantagens quando se emprega o mecanismo de captura das modificações da base de dados designado por *log pull*. O *log pull* continuamente executa um processo que extrai a informação das transacções terminadas (*commited*) a partir do *log* da base de dados e envia para o mecanismo de distribuição do serviço de replicação. Quando a metodologia de partição é usada com o sistema de gestão da base de dados relacional SGBD, as tabelas da base de dados são separadas em grupo. Um grupo contém tabelas ou dados que não são parte do processo de replicação. Esta partição diminui a quantidade de dados no *log* que o mecanismo *log pull* vai verificar. É uma boa maneira de particionar os dados quando ocorre

grande quantidade de actualização dentro da base de dados (*log* grandes) e somente poucas tabelas são assinaladas para a replicação.

A partição na fonte primária tem desvantagens comparado com a partição no secundário porque é complexa, complica a codificação das unidades distribuídas de trabalho e adiciona a complexidade nos cenários de recuperação. Isto quer dizer que as tabelas da base de dados podem não estar separadas quanto às restrições de integridade referencial, isto é, uma chave primária e todas as referências estrangeiras devem estar sempre dentro da mesma base de dados para a gestão das restrições de integridade referencial definidos.

Nas unidades distribuídas de trabalho, se uma réplica secundária precisa o resultado de uma unidade distribuída de trabalho, o mecanismo *log pull* pode não ser útil porque o processo de replicação não tem funcionalidades de reunir pedaços do mecanismo *log pull* separado. Os mecanismos de administração e recuperação são mais complexos. A recuperação de duas bases de dados com *logs* separados para o mesmo ponto no tempo é mais complexo que uma base de dados simples. Por isso, a partição na fonte primária não é recomendável para a empresa. A segunda alternativa é a partição na fonte secundária que traz vantagens nos cenários de recuperação da base de dados e um grau de transparência à acesso a dados. Uma escolha de partição no secundário reflecte uma perspectiva de replicação e administração da base de dados versos perspectiva de desenvolvimento da replicação. A partição de dados no secundário tem o impacto nos cenários recuperação da BD e no grau de transparência a acesso a dados.

6.9. Análise geral

A escolha de uma solução adequada depende dos custos da solução e do risco que representa. Todavia, nem todas as soluções apresentadas proporcionam de igual maneira os três elementos considerados principais neste estudo, embora a replicação em dois centros (Beira e Maputo) apresente uma solução equilibrada para todos. Para garantir uma tolerância a falhas e disponibilidade basta uma réplica num ponto geograficamente distinto do primário. Esta solução pode não ser suficientemente equilibrada para oferecer um melhor desempenho das aplicações e o equilíbrio da carga de dados. Nestas circunstâncias, uma replicação entre o Centro e o Norte pode ajudar, na medida em que cerca de 40% dos dados são de clientes do Centro e Norte, e pode não

exigir um servidor de capacidade igual ao primário em Maputo devido a menor número de actualizações. Uma réplica na Beira terá custos de administração e manutenção, mas vai proporcionar um equilíbrio nos acessos concorrentes de dados e pode melhorar a disponibilidade e o desempenho das aplicações.

Uma replicação utilizando servidor virtual resolve uma parte do problema a nível dos Sistemas de Informação que até agora não é muito relevante à empresa devido à falta de Sistema em pontos isolados e de forma arbitrária. Entretanto, uma disponibilidade contínua, está a emergir gradualmente na empresa devido a modernização dos processos de pagamento das facturas telefónicas por ATMs bancários. Este tipo de serviços vai gradualmente colocar outro tipo de desafios à empresa – os pagamentos electrónicos, que deveriam ser acompanhados de uma disponibilidade total e replicação de dados. Neste caso, a replicação heterogénea pode ser aproveitada como método de manutenção de consistência de dados recolhidos em fontes heterogéneas incluindo os fornecedores (ATMs) e clientes.

O fornecedor *Oracle* tem soluções de replicação de dados, apesar de a *Sybase* apresentar também soluções completas, tanto de *hardware* como de *software*. Não será portanto impossível redesenhar as aplicações actuais por forma a incorporarem a replicação de dados aqui proposta.

O problema de alta disponibilidade e tolerância a falhas foi tratado na base das tecnologias de bases de dados em particular a replicação. Mas um mecanismo seguro e efectivo de tratamento de tolerância a falhas e recuperação de desastres não existe na TDM. Até agora uma falha de *hardware* ou da base de dados requer em média 6 horas para reposição dos dados a partir das bandas magnéticas. Será que este tempo é aceitável? Os *backups* na TDM são responsáveis em parte do tempo em que as aplicações de produção estão indisponíveis. Além disso, estes *backups* não são simples, consomem o tempo e recursos humanos. Vários factores contribuem para a fraca capacidade no tratamento destes aspectos. Por exemplo, a tecnologia de armazenamento de dados baseia-se nos servidores monolíticos designados por *Direct-attached Storage (DAS)*. Esta tecnologia assenta nos SCSI dedicados e *Fibre Channel access path* que liga o servidor com os dispositivos de armazenamento de dados, incluindo os *drives* de discos e *arrays*, os *drives* de bandas magnéticas e as suas bibliotecas. As desvantagens destas tecnologias são:

a empresa não consegue adicionar ou remover novos recursos físicos do sistema sem interromper a exploração;

a tecnologia é limitada, a falha do subsistema (discos ou servidor) não permite as aplicações acederem os dados;

o mecanismo de recuperação de desastres é baseado somente nas bandas magnéticas *off-site* e não se recorre ao uso dos cofres.

o sistema *DAS* torna difícil a implementação de replicação dos dispositivos de armazenamento de dados devido ao fraco desempenho.

A solução efectiva destes problemas exige mais que uma camada de redundância (replicação) simples na empresa. Uma redundância com propósitos de protecção de dados (ver 6.6.1) deve garantir um bom desempenho na recuperação de desastres. Os métodos de recuperação de desastres estão relacionados com os mecanismos de armazenamento de dados e do tipo de *backup* (toda a base de dados, incremental ou diferencial). Um *backup* adequado para a TDM deve resolver o problema da partilha de dados em grandes subsistemas de arquivo de dados.

A recuperação de desastres pode ser tratada utilizando a tecnologia *Storage Area-Network* (SAN). A SAN tem maiores facilidades comparado com a tecnologia anterior. A SAN requer uma largura de banda suficiente (100MB/sec) para garantir um bom desempenho nas comunicações entre cliente/servidor. A grande vantagem de SAN assenta nas comunicações de muitos para muitos e a capacidade de maior conectividade e com bom desempenho, ver o anexo B. Segundo a *International Data Corp*. (IDC) um sistema é considerado de alta disponibilidade se, quando ocorre uma falha, não há perdas de dados e o sistema pode recuperar em tempo razoável [29]. O tempo considerado razoável depende do tipo de negócio. Um sistema de alta disponibilidade tem a disponibilidade igual a 99.99% [6,9]. Na TDM a disponibilidade das aplicações pode se considerar baixa. Em relação a probabilidade de falha das aplicações em estudo (tabela 6.1), também mostra que assume valores elevados comparados com os sistemas fiáveis, pois a probabilidade de um sistema funcionar 72 horas sem falhar é inferior a 95%.

Em relação a OF foi feito uma análise de uma amostra de 3 meses (Setembro a Novembro de 2002). Concluiu-se que o *upgrade* da aplicação exigia o *upgrade* de *hardware,* consequentemente a fiabilidade é de cerca de 74.019% e a disponibilidade quase igual ao do

estágio anterior. O valor da fiabilidade é ainda mais baixo comparado com o estágio anterior devido a prevalência das estratégias anteriores em relação à manutenção da base de dado e do *hardware*, *backups* e teste de novos produtos.

Capítulo VII

7. Conclusões

A presente dissertação visa melhorar três elementos chave dos Sistemas de Informação de uma grande empresa, nomeadamente tolerância a falha, disponibilidade e desempenho nos sistemas e bases de dados distribuídos.

A razão fundamental da abordagem prende-se com o facto de o candidato ao título académico estar a trabalhar nos SI da TDM, empresa em estudo, que pode enfrentar dificuldades a vários níveis pela falta de mecanismos que garantem os três elementos acima referidos. O conjunto destas preocupações constitui o problema de gestão enunciado nas primeiras páginas da dissertação.

O propósito do trabalho provém da grande preocupação de ver a TDM com um SI em que os serviços de base de dados sejam menos dependentes da infra-estrutura da rede, caracterizado de uma maior disponibilidade e desempenho. Essa preocupação levou a um estudo profundo do funcionamento do SI da empresa, de onde se tem como principais constatações:

- A centralização de recursos e dados num único ponto sem mecanismos de redundância num outro local separado da fonte primária, tornando todos os serviços da base de dados críticos e dependentes da infra-estrutura da rede.
- A baixa disponibilidade das aplicações Girafe, OF e Internet de aproximadamente 97.90%, 88.56% e 97.97% respectivamente, proveniente de várias interrupções quer planificadas quer acidentais.
- Um reduzido desempenho das aplicações visível num elevado tempo de resposta, com maior destaque nos pontos de acesso mais distantes como o Centro e Norte do País, para as aplicações Girafe, OF e serviços de Internet.
- Falta de um plano de recuperação de desastre com bom desempenho. As salvaguardas são feitas nas bandas magnéticas e são dispendiosos em relação ao tempo e recursos humanos.
- Os serviços de Internet por sua vez são dispendiosos e aumentam o trafego na rede uma vez que todos os utilizadores destes serviços dependem do centro de dados em Maputo onde estão os servidores primário e secundário. Esta constatação é ainda acrescida pela falta de uma

replicação via *cache* ou existência de um *proxy* nos locais chaves para tornar rápido o acesso ao serviço.

- Fraca padronização dos processos de entrada e saída de dados, leitura de bandas magnéticas das chamadas telefónicas e consultas incluindo impressão das facturas de telefone devido a falta de arquitectura que facilita a partilha de dados.

Posto isto, constituiu objectivo chave da dissertação a escolha dentre diversas técnicas de replicação de dados e serviços, à procura da mais apropriada às condições da empresa TDM por forma a conseguir um aumento significativo da tolerância a falhas, desempenho e segurança de dados nos sistemas e bases de dados distribuídos.
Utilizou-se a metodologia da pesquisa bibliográfica, diagnóstico e exame da tolerância a falhas, disponibilidade e desempenho para o Sistema de Informação da TDM, como estudo de caso real.

As amostras utilizadas para o cálculo quantitativo da disponibilidade e fiabilidade foram recolhidos durante o 1º semestre de 2002 nos *logfiles*, registos de *help desk* e observações feitas aos SI da empresa. Para uma melhor compreensão da análise qualitativa da disponibilidade e fiabilidade dos SI da TDM considerou-se, para efeitos de comparação, os sistemas analisados na bibliografia mais relevante [6,9,34,35]. Segundo a teoria dos "*Five-nines*" um sistema de alta disponibilidade tem uma interrupção de 32 segundos por ano, o que corresponde a 99.999% da disponibilidade. No estudo presente e para efeitos de comparação, foi considerado valor mais baixo de 3 dias 15 horas e 40 minutos por ano (tabela 7.1.) ou seja, 99%. Para a fiabilidade nesses estudos o valor recomendável é de 95% o que corresponde a probabilidade de falha de 5%.

Das várias conclusões deste estudo, há a realçar a baixa disponibilidade nas aplicações Girafe, OF e Internet. A aplicação OF apresenta valores mais baixos comparada com as outras aplicações. A menor disponibilidade nas aplicações Girafe e OF é originado pelas falhas e interrupções. As falhas englobam as falhas de energia, falhas do servidor ou da base de dados e da rede de comunicações de dados, enquanto que as interrupções incluem salvaguardas, manutenção, teste de novos produtos e facturação.

Aplicação	D(t)	Intervalo	Interrupção
R 1	99.000%	1 ano	3 dias 15 horas e 40 minutos
OF	88.560%	5 meses	16 dias 11 horas
*OF	89.940%	3 meses	7 dias 21 horas 33 minutos
Internet	97.970%	6 meses	3 dias 16 horas
Girafe	97.900%	6 meses	3 dias 19 horas 1 minuto

Tabela 7.1.Disponibilidade e duração das interrupções em dias

A fiabilidade é menor nas aplicações Girafe, OF e no servidor da base de dados de espera devido ao elevado número de falhas. As falhas que ocorrem são de energia, servidores ou bases de dados e a falha das comunicações. Nas áreas das TDM mais distantes Centro e Norte do País a probabilidade de falha é elevada comparada com o centro de dados porque a além das falhas já mencionadas acrescenta-se as falhas de energia no local, da rede local e da interface do cliente (PC). As implicações são a paralisação da actividade do utilizador e nalgumas situações alteração dos prazos de pagamento ou de cobrança de clientes e fornecedores.

A fiabilidade da base de dados de espera é menor comparada com as outras aplicações. A frequência da falha da base de dados de espera é aproximadamente, de quatro em quatro dias. As causas prováveis das falhas são entre outras o código que implementa a redundância e a insuficiência dos recursos do servidor, memória e processadores.

O desempenho foi avaliado sob o ponto de vista de tempo de resposta das aplicações concluindo-se que a latência dos canais de comunicação afecta seriamente os locais remotos do Centro e Norte do País.

A solução recomendada para melhorar os três elementos em estudo consiste na utilização de replicação passiva em dois centros de dados localizados em Maputo e na Beira, utilizando a tecnologia de arquivo de dados SAN. A proposta de uma arquitectura de replicação em dois locais independentes visa tornar os Sistemas de Informação da TDM disponíveis e tolerantes a falhas (fiáveis). Pretende-se evitar ou minimizar que se uma componente do sistema falha, seja servidor de dados ou canal de comunicação, todo o sistema pare. Em termos práticos, só é possível utilizando sistemas de processamento paralelo com falhas independentes, e no caso da TDM é recomendável a utilização da replicação de parte do sistema. A proposta apresenta garantia da independência das falhas através de separação das instalações do *hardware*, isolamento de redes de energia e de serviços de telecomunicações.

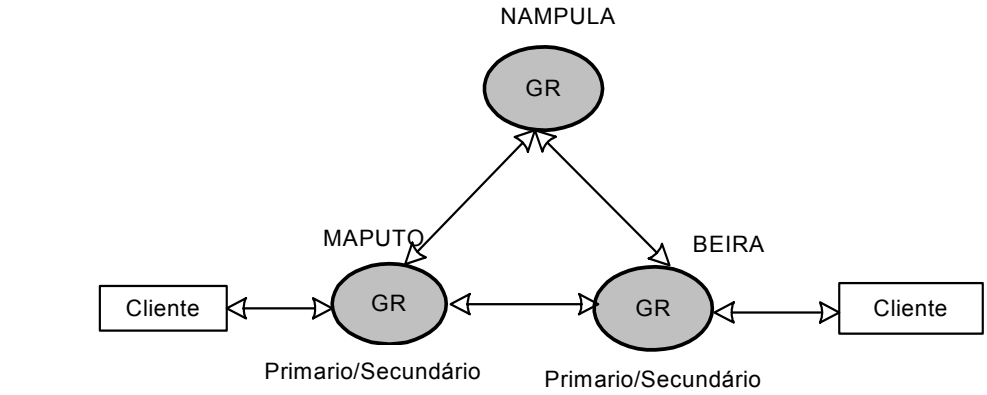


Figura 7.1. Arquitectura da replicação

A figura 7.1 representa esquematicamente a arquitectura de implementação em três locais críticos. No esquema, Maputo e Beira terão Gestores de dados, os servidores. Cada servidor será fonte primária de dados nesse local e fonte secundária em relação ao outro. Quer dizer, a replicação é passiva e os intervalos de actualização do estado dos dois servidores de dados nos centros de dados será definido em coordenação com a gestão da empresa. Em Nampula, não é necessário um servidor de dados, mas sim uma redundância dos meios de comunicação para o equilíbrio do trafego e bom desempenho.

As vantagens desta arquitectura são: a melhor disponibilidade e desempenho tanto dos servidores como da rede de dados. A empresa passa a dispor de um plano de recuperação de base de dados e desastres fiável e eficiente (bom desempenho) em comparação com o mecanismo baseado nas bandas magnéticas uma vez que cada centro de dados vai formar um grupo de recuperação de dados, as salvaguardas e restaurações serão planificadas como parte destes.

A utilização da tecnologia de arquivo de dados SAN traz um custo efectivo na expansão do negócio e partilha de dados. A integração de novos locais na WAN incluindo novos produtos não vai implicar novos investimentos para aumentar as facilidades e capacidades no *hardware* incluindo o custo de deslocamento e *hardware* de ligação.

A arquitectura da rede de dados proposto consiste na redundância de canais de comunicação nos pontos críticos e uma replicação via cache. Esta proposta procurou tomar em consideração o enquadramento das tecnologias das aplicações, o tempo de recuperação da falha, custos e a facilidade de utilização incluindo a configuração e manutenção. Por estas razões, a replicação dos canais consistiu em três locais estratégicos Maputo, Beira e Nampula que correspondem a zona Sul, Centro e Norte do País respectivamente. As aplicações devem ter um mecanismo de acesso via *web*.

Em relação as salvaguardas de dados na empresa, uma cópia na Beira constitui por si só uma salvaguarda e ajuda na diminuição de custos, na optimização do desempenho de todo o sistema, eliminando as salvaguardas diários nas bandas magnéticas. Se for implementada uma replicação passiva com a técnica dos *snapshots*, torna ainda mais fácil os procedimentos de *backup* com menor impacto nas aplicações de produção uma vez que é possível isolar um *snapshot* e realizar uma salvaguarda sem interromper todos os utilizadores. Esta estratégia de *backup* toma em consideração o volume das transacções em cada local. No caso da TDM a interrupção do *snapshot* com dados do Centro e Norte do País tem menor impacto devido ao menor número das transacções que representa, aproximadamente 40% de total das transacções. As estratégias de salvaguarda de dados podem ser automatizados desde que a TDM inclua como requisito junto aos fornecedores no processo da compra da tecnologia recomendada SAN.

Os requisitos necessários para implementar a arquitectura de replicação passiva Maputo/Beira são dois. O primeiro corresponde à alteração do sistema de arquivo de dados em vigor na empresa. A tecnologia DAS torna difícil a replicação e partilha de dados em grandes subsistemas e no seu lugar propõe-se a tecnologia SAN. O segundo requisito é aumento da largura da banda de 64/128 *Kbits* para 100 *Mbits* por segundo incluindo toda a interface devido ao elevado volume de dados em circulação na rede que podem congestionar o canal e originar um fraco desempenho.

A solução proposta acarreta elevados custos para a empresa em particular na aquisição de novo *hardware* e *software* de replicação, que para a tecnologia SAN poderá atingir o dobro dos valores da tecnologia DAS.

Uma alternativa económica é implementar a replicação passiva Maputo/Beira utilizando um sistema misto mantendo a parte do sistema de arquivo DAS já existente na TDM e a

adquirindo um sistema de arquivo de dados SAN para o centro de dados em Maputo para dar resposta ao elevado número das transacções.

Conforme já foi referido anteriormente, recomenda-se também nesta solução o aumento da largura de banda para 100 *Mbits* por segundo, incluindo interfaces para suportar o volume de dados em circulação na rede e evitar o congestionamento. O reaproveitamento da infra-estrutura é avaliado como uma das vantagens comparativas para a implementação da solução.

A implementação da replicação na TDM implica custos de aquisição de *hardware* e *software* de replicação. A replicação de dados e serviços implica a duplicação de esforços nas operações de actualização exigindo da empresa uma maior largura da banda. Para além destes custos, uma replicação pode dar origem a um congestionamento e sobrecarga do sistema o qual vai exigir da empresa sistemas adicionais para detectar este tipo de falhas. Uma replicação tem custos elevados no sistema de armazenamento e dispositivos de entrada e saída de dados porque exigem uma largura de banda a altura da replicação. É recomendável ainda a formação de um técnico de manutenção e configuração de *hardware* das comunicações na zona Centro e Norte do País. Para o centro de dados na Beira, é necessário uma equipa de manutenção e administração dos serviços elementares da base de dados e rede.

Apesar dos custos acrescidos, a replicação é indispensável face aos indicadores de disponibilidade e desempenho recolhidos. Não é aceitável que na TDM serviços chave em média não consigam operar mais de 3 dias sem sofrerem uma paragem.

Esta dissertação é ponto de partida para diversas propostas de desenvolvimento dos SI da TDM, razão pela qual existe muito trabalho futuro. Um dos aspectos a melhorar e que muito poderiam beneficiar o trabalho aqui realizado é o registo preciso de todas as falhas e dos locais e componentes que falharam. A sua ausência impede, por exemplo, uma análise de falhas que inclua regressões lineares e tendências de modo a permitir identificar o grau de contribuição de cada factor.

Em relação a disponibilidade e confiabilidade uma boa análise seria considerar uma amostra de um ano conforme se recomenda na literatura.

Apesar das dificuldades encontradas na obtenção de indicadores de disponibilidade e desempenho, o autor acredita que a análise aqui apresentada bem como a discussão em torno das alternativas para o modelo de replicação de dados e serviços da TDM constituem um contributo

valioso para a melhoria do funcionamento dos Sistemas de Informação desta empresa. Por conseguinte, os objectivos inicialmente propostos para este trabalho terão sido alcançados.

Anexo A

A disponibilidade e a fiabilidade foram calculadas com base nas fórmulas de disponibilidade *D(t)* e fiabilidade *C(t)* do capítulo II do relatório. A partir da função de distribuição $f(x) = \alpha e^{-\alpha x}$ calcula-se o coeficiente $\alpha = \frac{1}{\bar{x}}$

sendo $\bar{x} = \frac{\sum Ui}{\sum fi}$ onde *Ui* é o tempo total de funcionamento sem falhas e *fi* frequência de falhas.

A partir da função de fiabilidade *C(t)* obtém-se as funções particulares de cada aplicação. Em todas as análises foi considerado um intervalo de quatro dias numa escala de 6 em 6 horas. A escolha de 72 horas para o cálculo de fiabilidade tal como foi referido no capitulo 6 foi o resultado deste valor oferecer uma análise mais significativa do valor da fiabilidade e do intervalo de tempo necessário para o funcionamento de um sistema. Isto quer dizer que, considerando o tempo superior a 72 horas, a fiabilidade e a probabilidade de falhas apresentam valores muito baixos e conduz a uma análise pessimista dos Sistemas de Informação. Se se considerar um intervalo inferior a 72 horas resulta numa análise optimista que não se enquadra neste estudo porque o tempo de funcionamento é menor a fiabilidade é maior para sistemas que deviam funcionar todos os dias sem parar.

A tabela a seguir apresenta as funções de distribuição das falhas das aplicações a partir das quais foram calculados os valores quantitativos da fiabilidade.

Item	Aplicação	Função da fiabilidade	Função da probabilidade de falhas
1	OF	$C(t) = e^{-\frac{t}{552.02}}$	$P(t) = 1 - e^{-\frac{t}{552.02}}$
	OF (3 meses)	$C(t) = e^{-\frac{t}{239.33}}$	$C(t) = 1 - e^{-\frac{t}{239.33}}$
2	Girafe	$C(t) = e^{-\frac{t}{430.46}}$	$C(t) = 1 - e^{-\frac{t}{430.46}}$
3	Internet	$C(t) = e^{-\frac{t}{2135.5}}$	$C(t) = 1 - e^{-\frac{t}{2135.5}}$
4	*Standby Database*	$C(t) = e^{-\frac{t}{132.92}}$	$C(t) = 1 - e^{-\frac{t}{132.92}}$

Tabela A.1. Funções de distribuição

Estas funções exponenciais de distribuição de falhas foram obtidas a partir das amostras das falhas dos sistemas apresentados na tabela A.2.

Amostra de 1º Semestre de 2002

Internet

No.	Data	Duração(horas)
1	15-Mar-02	69
2	Apr-02	4

OF

No.	Data	Duração(Horas)
1	09-Jan-02	24.55
2	15-Jan-02	16.6
3	13-Feb-02	21.51
4	05-Mar-02	14.83
5	24-Apr-02	23.91
6	08-May-02	18.45

Girafe

No.	Data	Duração(Horas)
1	11-Feb-02	2.08
2	23-Feb-02	1.23
3	24-Feb-02	6.18
4	16-Feb-02	7.61
5	28-Feb-02	0.33
6	08-Mar-02	7.06
7	24-Apr-02	1.71
8	07-May-02	8.63
9	27-May-02	3.77
10	10-Jun-02	0.7

Tabela A.2. Amostra de falhas não planificadas em A

Amostra de OF

Falha	Hora	Recuperação	Hora	Duração	Motivo
4-Sep-02	18:48:00	5-Sep-02	8:17:00	13.483	*Backup*
5-Sep-02	10:21:00	5-Sep-02	11:52:00	1.5	Falha
11-Sep-02	18:21:00	12-Sep-02	8:54:00	14.55	*Backup*
13-Sep-02	11:32:00	13-Sep-02	17:05:00	5.55	Falha da BD
17-Sep-02	16:03:00	17-Sep-02	20:43:00	4.667	Inter. não esclarecida
17-Sep-02	22:00:00	17-Sep-02	22:43:00	0.71	Falha da BD
18-Sep-02	0:52:00	18-Sep-02	9:48:00	10	Falha do servidor
20-Sep-02	17:02:00	23-Sep-02	7:30:00	14.5	*Backup* e fecho do semestre
2-Oct-02	18:11:00	3-Oct-02	7:51:00	13,667	*Backup*
8-Oct-02	10:19:00	8-Oct-02	11:32:00	1.2167	Falha de energia
3-Oct-02	7:51:00	3-Oct-02	11:31:00	3.667	Falha da BD
3-Oct-02	14:01:00	3-Oct-02	14:30:00	0.5	Falha de energia
3-Oct-02	17:38:00	3-Oct-02	19:00:00	1.367	Manutenção de energia
11-Oct-02	17:28:00	12-Oct-02	7:51:00	14.383	*Backup*
12-Oct-02	8:31:00	12-Oct-02	11:54:00	3.383	Falha
14-Oct-02	15:48:00	14-Oct-02	16:08:00	0.33	*Shutdown/Startup*
18-Oct-02	19:13:00	19-Oct-02	8:05:00	12.867	*Backup* da BB
25-Oct-02	19:00:00	26-Oct-02	9:05:00	14.083	*Backup* mensal e falha (1.5 h)
30-Oct-02	19:00:00	31-Oct-02	7:27:00	12.5	Backup BD
6-Nov-02	18:07:00	7-Nov-02	7:58:00	13.85	Manutenção
13-Nov-02	18:07:00	14-Nov-02	7:00:00	12.883	*Backup* mensal
16-Nov-02	12:36:00	16-Nov-02	17:48:00	5.2	*Shutdown/Startup*
20-Nov-02	18:26:00	21-Nov-02	7:34:00	13.13	*Backup* semanal
21-Nov-02	7:48:00	21-Nov-02	11:17:00	3.48	Falha da BD
29-Nov-02	17:33:00	30-Nov-02	7:09:00	13.6	*Backup* Mensal

Tabela A.3. Amostra de um trimestre

Função de distribuição de falha $P(t) = 1 - e^{-\frac{t}{239.33}}$

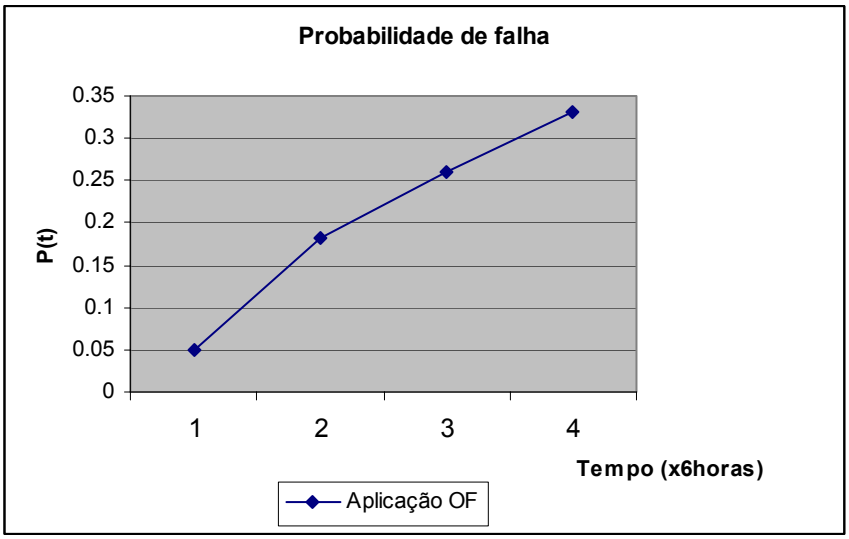


Figura A.1. Gráfico da probabilidade da falha da aplicação OF

Anexo B

Um sistema de armazenamento de dados deve satisfazer os requisitos do negócio e as necessidades dos utilizadores. As necessidades dos utilizadores incluem a disponibilidade e o desempenho. Os requisitos do negócio incluem um ambiente flexível e reutilizável que acompanha o crescimento do negócio, a continuidade das operações, integração de dados e aplicação para o suporte à decisão e planificação. O negócio exige também um mecanismo confiável de segurança de dados, os *backups*. Um *backup* deve existir numa empresa e deve ser protegido. Um *backup* requer uma planificação das estratégias da sua realização e deve ter um menor impacto nas aplicações de produção. Uma solução efectiva do *backup* requer mais do que uma camada de redundância num meio simples.

Actualmente grande parte das organizações guardam os seus dados nas bandas magnéticas que garantem uma protecção de dados. O mecanismos das bandas magnéticas evoluiu mas a sua escolha é um exercício necessário e de responsabilidade porque nalguns casos prevalecem as questões de desempenho e disponibilidade. Além disso, uma planificação cuidadosa do *backup* nas bandas magnéticas vai permitir a identificação do tipo de dados ou ficheiros que possam ser guardados nas bandas e quais são acedidos imediatamente a partir dos *arrays* dos discos, que tipo de *backup,* total, incremental ou diferencial.

Uma *backup* adequado deve resolver problemas de partilha de dados em grandes subsistemas de discos para uma melhor utilização, deve incluir o *mirrorring* de *arrays* redundantes e ter, se possível, armazenamento de dados distribuídos incluindo os *backups* com uma gestão centralizada. Segundo estudos de Índices de vulnerabilidade 82% das companhias/corporações estabelecidas em todo o mundo não tem uma protecção adequada e efectiva contra problemas de remoção acidental, corrupção de dados, falhas do sistema, vírus ou desastres do meio envolvente [29,30].

Na TDM

As aplicações da empresa são independentes e requerem uma integração para permitir a partilha de dados importantes. Em relação aos dispositivos de armazenamento de dados é necessário uma tecnologia capaz de permitir a adição de subsistemas de armazenamento de dados

sem interromper os processos do negócio. A gestão dos arquivos de dados deve ser eficiente e evitar desperdício do espaço. A empresa TDM utiliza a tecnologia *Direct-Attached Storage* (DAS) com *mirror* dos discos. Esta tecnologia oferece uma privacidade dos dispositivos de armazenamento de dados, mas não permite o isolamento das falhas entre os subsistemas dos discos e servidor, afectando tanto a disponibilidade como o desempenho. O processo de recuperação de dados incluindo desastres na TDM consiste no mecanismos *offline,* vulgarmente conhecido por mecanismo das bandas magnéticas de recuperação.

Os *backups* dos dados da empresa e processos de recuperação ocorrem num único ponto, no centro de dados, e não existe a replicação de dados. Para além disso, a nível da empresa é necessário diferentes ferramentas de gestão e administração de cada ambiente do servidor tornando o sistema complicado. A adição e configuração ou reconfiguração das capacidades dos discos é uma tarefa cara e ineficiente tal como o desempenho e a monitorização do sistema. Num sistema DAS a partilha de dados depende dos subsistema de discos e de um servidor específico. Qualquer falha num dos subsistemas torna os dados indisponíveis. E a partilha de dados entre diferentes aplicações na rede pode dar lugar um considerável efeito no desempenho [29,30].

As soluções alternativas para a empresa passam pela mudança da tecnologia de armazenamento de dados. Existem duas tecnologias alternativas de armazenamento de dados a *Network Area Storage* (NAS) e a *Storage Area Network* (SAN), que diferem entre si no modelo de arquivo (centralizado ou distribuído) e nos custos. A tecnologia NAS oferece ao cliente e grupo de trabalho da TDM acesso directo aos dados. Mas, o NAS tem as mesmas limitações que apresenta o DAS em particular a alta disponibilidade e o desempenho, a falha de um subsistema de discos torna impossível o acesso à dados. Um *backup* na tecnologia NAS tem impacto no domínio do cliente/servidor por causa do dispositivo *thin server* que está num local dentro da LAN exigindo uma largura de banda e afectando o desempenho dos recursos de comunicação entre o cliente e o servidor. A terceira solução é a utilização da tecnologia SAN que corresponde ao novo paradigma dos dispositivos de armazenamento de dados com custo efectivo. A SAN implementa os *arrays* modulares e biblioteca das bandas magnéticas como alternativas de *arrays* centralizados. Uma SAN aumenta a conectividade entre o servidor e *arrays* centralizados, biblioteca das bandas magnéticas amortizando os custos de armazenamento num grande número de servidores. A SAN implementa a comunicação de muitos para muitos em vários servidores e

dispositivos de armazenamento. A tecnologia SAN permite uma partilha de diferentes dispositivos de armazenamento de dados podendo ser acedidos por diferentes servidores, e facilita a replicação de dados entre os diferentes dispositivos de armazenamento de dados. O acesso a dados tem um bom desempenho porque implementa vários caminhos de acesso a dados resultando numa elevada disponibilidade e desempenho.

Os *backups* a nível da empresa podem ser simplificados porque a tecnologia SAN permite incorporar as bibliotecas das bandas magnéticas. A gestão do crescimento é mais fácil podendo ser feito a partir de um ponto central e com uma ferramenta simples. A SAN tem facilidades para remoção, adição e reconfiguração de dispositivos de modo transparente e flexível. A outra vantagem da SAN consiste na facilidade de implementar os *clusters* de alta disponibilidade que podem se estender até duzentos servidores e dispositivos de armazenamento de dados. A falha do servidor não afecta o acesso a dados. No estágio actual (DAS) quando um disco ou controlador do disco falha o sistema redundante gera um alarme. Quando a componente que falhou não for reparado ou substituído, a componente redundante também pode falhar, paralisando todo o sistema.

A SAN permite uma administração eficiente centralizada a partir de uma consola. As tarefas administrativas podem ser centralmente geridas, baixando os custos de gestão de armazenamento de dados e aumentar significativamente a consistência dos dados. A SAN reside entre o *host*, adaptador *bus* e o dispositivo de armazenamento. Esta posição cria pontos críticos no nível físico. Estes pontos eliminam pontos simples de falha implementando vários caminhos e infra-estrutura de dispositivos redundantes. Uma largura de banda máxima é um requisito importante devido ao elevado volume de dados a transferir. As ligações baseadas em *fibre channel* em uso na empresa podem suportar até 100MB/sec e uma elevada taxa de transferencia para o acesso rápido.

Em resumo a solução SAN é vantajoso para a empresa devido as facilidades na recuperação de desastres, gestão de crescimento, partilha de dados e alta disponibilidade. A figura B.1 mostra a implementação de uma SAN em dois centros de dados na TDM. Os discos centralizados podem permitir o armazenamento de dados heterogéneos, Windows NT, Unix e Oracle. Algumas considerações nesta solução dizem respeito à largura de banda e a

compatibilidade de todo o hardware envolvido incluindo os interfaces (*NICs*) pois empurrar 100 MB/*sec* numa SAN pode dar lugar a uma sobrecarga e congestionamento.

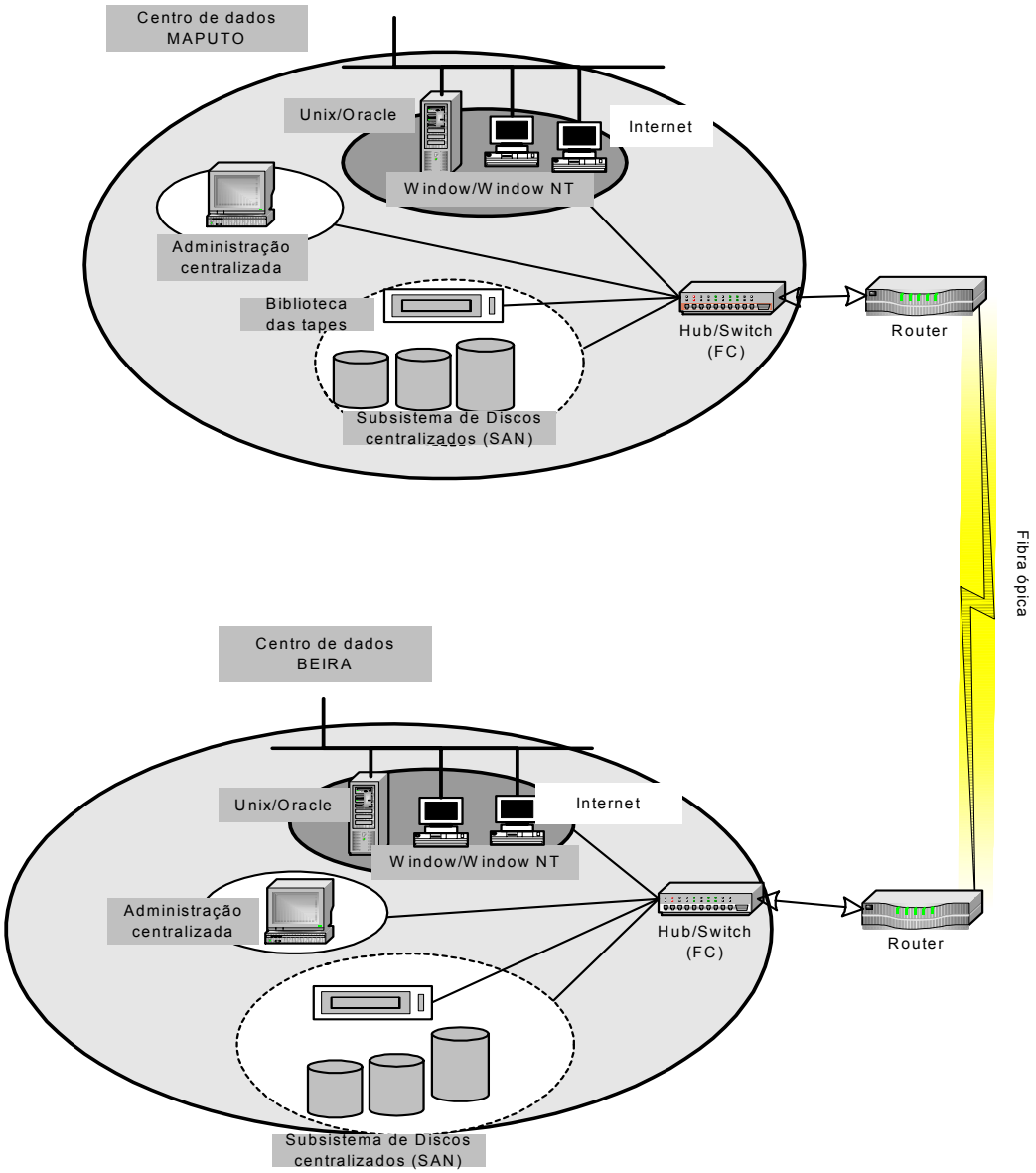


Figura B.1. Replicação de duas SAN´s

Mas, esta solução é dispendiosa para a empresa requerendo uma alteração de todo o sistema de armazenamento de dados. O custo de um sistema de armazenamento de dados NAS é metade do preço de uma solução SAN. Para a replicação de dados com objectivo de alcançar um bom

desempenho, tolerância a falhas e alta disponibilidade incluindo a recuperação de desastres uma solução mista constituído por uma SAN e dois DAS, pode ser económico para a TDM. Porque esta solução permite o reaproveitamento do sistema antigo e adicionar uma SAN. Quer dizer, a solução referida em 6.6.1 e 6.6.2 podem se fundir passando a ter um servidor na Beira equipado com a tecnologia DAS e um outro em Maputo também com a tecnologia DAS e uma SAN na Matola ou um outro local diferente do centro de dados. Um servidor na Beira e Maputo funcionarão como *cluster* de alta disponibilidade e equilíbrio da carga computacional (desempenho). Enquanto que a SAN na Matola (ou outro local) pode ajudar na partilha de dados crescimento do negócio e outras facilidades incluindo os *backups,* a figura B.2 mostra a implementação da replicação nos dispositivos de armazenamento de dados com tecnologia mista.

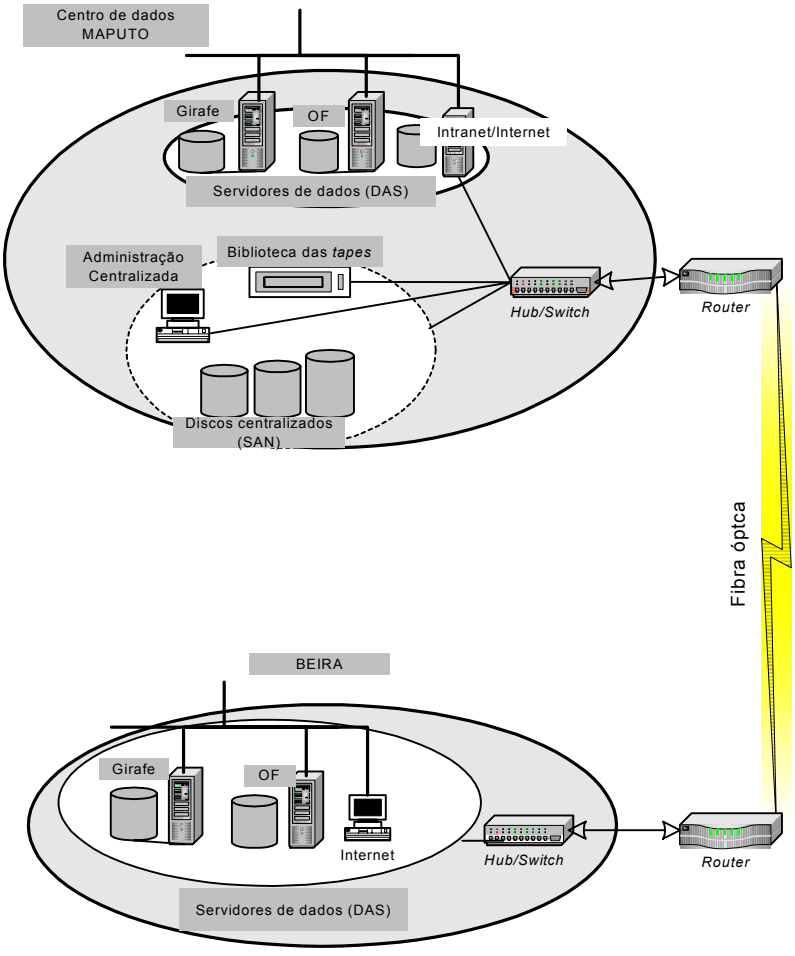


Figura B.2. Replicação mista

Referências Bibliográficas

[1] *Abdelsalam A. Helal, Abdelsalam A. Heddaya, Bharat B. Bhargava*, *Replication Techniques in Distributed Systems*, *USA, Kluwer Academic Publisher, 1996*

[2] *Elmasri/Navathe, Fundamentals of Database, 2nd Edition, USA, The Benjamin/Cummings Publishing Company, Inc. 1994*

[3] *Thomas Connolly and Carolyn Begg, Database system, 2nd Edition, UK, Addison-Wesley, 1999*

[4] *Lave Singh, Kelly Leigh, Joe Zafian et al. Oracle 7.3 Developer's Guide, 1ª Edição, USA, SAMS Publishing, 1997*

[5] Sape Mullender, Distributed System, 2nd Edition, New York, Addison-Wesley, 1994

[6] Andrew S. Tanenbaum, Distributed Operating Systems, USA, Prentice-Hall, Inc. 1995

[7] José A Marque e Paulo Guedes, Tecnologia de Sistemas Distribuídos, 1ª Edição, Portugal, FCA Editora de Informática, 1998

[8] Pradeep K. Sinha, Distributed Operating Systems, New York, IEEE Computer Society Press, 1997

[9] Coulouris Dollimore Kindberg, Distributed System, 3trd Edition, UK, Addison Wesley, 2001

[10] José Luís Pereira, Tecnologia de Bases de dados, 2ª Edição, Portugal, FCA - Editora de Informática, 1997

[11] C. J. Date, Database system, 7th Edition, EUA, Addison-Wesley, 2000

[12] Site http://dcc.ing.puc.cl/~jnavarro/iic2332/apuntes/apuntes_11.html

[13] Nikhil Chandhok, Web Distribuition System, http://www.cis.ohio-state.edu/~jain/cis788-99/web_caching/index.html, Computer Networking and Telecommunications Research Lab, Ohio Columbus, 7/2000

[14] Pete Loshin, TCP/IP Clearly Explained, 2nd Edition, USA, Academic Press, 1997

[15] Coulouris Dollimore Kindberg, Distributed System, 2nd Edition, UK, Addison-Wesley, 1994

[16] Nancy Thalia, Reynolds et al., Networking Essential, 3rd edition, USA, Microsoft Press, 2000

[17] J.S. Milton, Jesse C. Arnold, Introduction to Probability and Statistics, 3rd Edition, New York McGraw-Hill, Inc, 1995

[18] David Bell, Jane Grimson, Distributed Database systems, UK, Addison-Wesley, 1992

[19] Marie Buretta, Data Replication: Tools and Techniques for Managing Distributed Information, New York, Wiley Computer Publishing, 1997

[20] Paulo Veríssimo e Luís Rodrigues, Distributed System for System Archtects, Portugal, Kluwer Academic Publisher, 2001

[21] Phillip Russion, Strategies and Syabase Solution for Database Availability,EUA, Hurwitz Group Inc. 11/2001

[22] http://www.ibm.com

[23] Bob Thome, The High Availability Database Server Cookbook, http://otn.oracle.com/deploy/availability/pdf/oow_273_ha_db_server_cookbook.pdf, Oracle corporation, 2001

[24] Rachid Guerraoui, André Schiper, Fault-Tolerance by Replication in Distributed Systems, Switzerland

[25] http://www.raid-advisory.com/rabguide.html

[26] Sybase, Disaster Recovery, USA, http://sybase.com, November 2001

[27] Dominic J. Delmolino, Strategies and Techniques for using Oracle 7 Replication, USA, Oracle Corporation, 1995

[28] TDM, Relatório de contas 2001, Moçambique 2001

[29] Hp, Answering the Challenges of Enterprise Storage, , Hewlet-Packard Company, 2001

[30] Rob Strechay, Implementing A SAN, USA, Business Communication Review, Aug 2002

[31] Mathias Wiesmann, Fernando Pedone, André Schipher et al., Database Replication Techniques: a three parameter classification, Germany, IEEE Computer Society , October 2000

[32] Zahir Tari Omran Bukhres, Fundamental of Distributed Object System The Corba Perspective, USA, Wiley – Interscience Publication John Wiley & sons, Inc, 2001

[33] Rondolph A Fisher, CBCP, Reliability Options for High-Speed Packet Data Network, www.webtutorial.com, 2002

[34] Stevan Taylor, Evaluating and Calculating "Five Nines", Editor Webtorials.com, 2003 http://www.webtorials.com/main/eduweb/manfr99/tutorial/five-nines/five-nines.pdf

[35] Gary Audin, Reality Check On Five-Nines

http://www.bcr.com/bcrmag/2002/05/p22.asp, 2000